

UNIVERSITATEA NAȚIONALĂ DE ȘTIINȚĂ ȘI TEHNOLOGIE POLITEHNICA  
BUCUREȘTI

ȘCOALA DOCTORALĂ INGINERIA SISTEMELOR BIOTEHNICE

# REZUMATUL TEZEI DE DOCTORAT

**Contribuții asupra algoritmilor de indicare a mișcării  
pentru simulatoare de zbor**

**Autor:** Ing. Alina-Ioana CHIRA

**Conducător de doctorat:** Prof. dr. ing. Ion STROE

## COMISIA DE DOCTORAT

Președinte	Prof dr. ing. Andrei CRAIFALEANU	de la	Universitatea Națională de Știință și Tehnologie Politehnica din București
Conducător de doctorat	Prof. dr. ing. CS I Ion STROE	de la	Universitatea Națională de Știință și Tehnologie Politehnica din București
Referent	CS I dr. ing. Cătălin NAE	de la	Institutul Național de Cercetare Dezvoltare Aerospațială „Elie Carafoli” - INCAS
Referent	CS I dr. ing. Constantin OLIVOTTO	de la	Academia Tehnică Militară Ferdinand I
Referent	Prof. dr. ing. Adrian- Mihail STOICA	de la	Universitatea Națională de Știință și Tehnologie Politehnica din București

București  
2024

## Cuprins

1. Introducere .....	4
1.1. Context .....	4
1.2. Motivație .....	5
1.3. Obiectivele cercetării.....	5
1.4. Structura tezei.....	5
2. Stadiul actual al cercetării privind modelarea matematică a algoritmilor de mișcare specifici simulatoarelor de zbor .....	6
2.1. Introducere în problematica algoritmilor de indicare a mișcării pentru simulatoare de zbor	6
2.2. Algoritm filtru de constrângere clasic .....	6
2.3. Algoritm filtru de constrângere control optimal .....	7
2.4. Centralizarea caracteristicilor algoritmilor de indicare a mișcării .....	7
3. Modelare matematică a unui sistem de simulare dinamică pentru aplicații de tip pilot-in-the-loop .....	10
3.1. Introducere – prezentare generală a sistemului de simulare dinamică.....	10
3.2. Sisteme de coordonate.....	12
3.2.1. Sistem de referință avion .....	12
3.2.2. Sistem de referință simulator .....	12
3.2.3. Sistem de referință pilot-avion.....	13
3.2.4. Sistem de referință pilot-simulator.....	13
3.2.5. Sistem de referință inerțial .....	13
3.3. Transformări de coordonate .....	13
3.4. Modelarea matematică a sistemului robotic serial .....	14
3.4.1. Cinematica directă a robotului serial .....	14
3.4.2. Cinematica inversă a robotului serial.....	16
4. Scalarea spațiului de lucru pentru sistemul de simulare dinamică .....	18
4.1. Limitări hardware ale unghiurilor articulațiilor .....	19
4.2. Eliminarea coliziunilor.....	20
4.3. Limitare software ale unghiurilor articulațiilor .....	20
5. Modelarea matematică a algoritmului de indicare a mișcării .....	22
5.1. Noțiuni introductive .....	22
5.2. Scalare și limitare .....	23
5.3. Modelul matematic al sistemului vestibular.....	25

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

5.4. Procedura de implementare a algoritmului de indicare a mișcării .....	29
5.5. Algoritm genetic.....	30
6. Rezultate .....	35
7. Concluzii și dezvoltări ulterioare .....	42
7.1. Concluzii generale.....	42
7.2. Contribuții proprii .....	44
7.3. Perspective de dezvoltare a activității de cercetare .....	45
Bibliografie (selectivă).....	46

## 1. Introducere

Lucrarea de doctorat propune dezvoltarea unui algoritm de indicare a mișcării pentru un simulator de zbor bazat pe o platformă de mișcare de tip robot serial fiind utilizați algoritmi genetici pentru optimizarea filtrului de constrângere optimal bazat pe regulator liniar pătratic (LQR). Proiectarea, implementarea și testarea algoritmului a fost realizată utilizând sistemul de simulare dinamică pentru aplicații de tip pilot-in-the-loop, RoFSim, dezvoltat la Institutul Național de Cercetare-Dezvoltare Aerospațială „Elie Carafoli” - INCAS. Algoritmul de indicare a mișcării propus convertește dinamica aeronavei în mișcările robotului, generând mișcări de înaltă fidelitate în limitele fizice ale simulatorului de zbor și reproducând forțe și accelerații specifice, astfel putând fi simulate senzații similare celor experimentate într-o cabină reală de avion.

Tema de cercetare propusă este necesară în contextul în care crearea unui mediu realist de simulare a zborului aeronavelor existente și în curs de dezvoltare este obligatorie pentru validarea de scenarii de zbor, noi tehnologii și pentru antrenarea piloților pentru situații ce pot apărea în timpul operării aparatelor de zbor.

Rezultatele obținute demonstrează stabilitatea și buna funcționare a algoritmului filtru de constrângere optimal implementat și posibilitatea utilizării algoritmului propus în proiecte de cercetare-dezvoltare și dezvoltarea ulterioară a acestuia.

Cuvinte cheie: *simulator de zbor, algoritm de indicare a mișcării, sistem robotic serial, filtru de constrângere optimal, LQR, algoritm genetic, sistem vestibular*

### 1.1. Context

Simularea zborului a fost mult timp un instrument de importanță vitală în domeniul aerospațial. Pentru a minimiza riscurile și costurile de dezvoltare a aeronavelor ([1], [2], [3], [4]) au fost create medii de simulare a zborurilor cu scopul de a îmbunătăți calitatea acestora [5], menținând în același timp un nivel constant de siguranță operațională. Cercetătorii utilizează intensiv simulatoarele de zbor, pentru a obține informații în domenii precum procedurile de pregătire, înțelegerea limitelor interfeței pilot/avion și ergonomie. În acest context au fost dezvoltate simulatoare cu platformă fixă și simulatoare bazate pe platforme cu mișcare, roboți paraleli și roboți seriali.

În ultimele decenii, sarcina utilă a sistemelor robotice comerciale cu șase axe independente a crescut în mod sistematic. Primele simulatoare de zbor bazate pe sisteme robotice seriale, dezvoltate în scopuri de cercetare, au fost Diamond DA42 de la Centrul Aerospațial German (DLR) ([6], [7]) și Cyber-Motion Simulator de la Institutul Max Planck (MPI) [8].

Un efort în această direcție îl reprezintă sistemul de simulare dinamică pentru aplicații de tip pilot-in-the-loop, numit RoFSim ([9], [10], [11]), un simulator de zbor ce utilizează ca platformă de mișcare un robot serial. Acest simulator a fost dezvoltat la Institutul Național de Cercetare-Dezvoltare Aerospațială „Elie Carafoli” - INCAS. Simulatorul oferă capacități de indicare a mișcărilor pe lângă funcțiile de bază ale oricărui simulator de zbor. În timp ce performanțele hardware ale sistemului de mișcare al simulatorului de zbor au avansat substanțial, dezvoltarea algoritmului de indicare a mișcărilor care transformă dinamica aeronavei simulate în comenzi de mișcare realizabile, este ușor întârziată. Astfel, în cadrul

# Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

acestei teze de doctorat este formulat, proiectat, dezvoltat și implementat un algoritm optim de indicare a mișcării care convertește dinamica aeronavei în mișcările robotului, generând mișcări de înaltă fidelitate în limitele fizice ale simulatorului de zbor reproducând forțe și accelerații specifice, astfel putând fi simulate senzații similare celor experimentate într-o cabină reală de avion.

## 1.2. Motivație

Întrucât sistemul de simulare dinamică pentru aplicații de tip pilot-in-the-loop este un simulator utilizat în scopuri de cercetare ([11]), lucrarea își propune proiectarea unui algoritm optim de indicare a mișcării care poate transforma mișcarea reală a unei manevre de simulator de zbor într-o mișcare logică și ușor de înțeles pentru spațiul de lucru al robotului serial.

## 1.3. Obiectivele cercetării

Obiectivul general stabilit în lucrarea de doctorat este de a îmbunătăți sistemul de simulare dinamică pentru aplicații de tip pilot-in-the-loop în moduri semnificative care să-i sporească utilitatea prin creșterea nivelului de realism și a ușurinței de utilizare. Având în vedere acest lucru, un algoritm optim de indicare a mișcării a fost formulat pentru îmbunătățirea mediului de simulare a zborului, util în analizele de stabilitate ale noilor configurații de aeronave [12], bazate pe arhitecturi de tip electric- hibrid.

## 1.4. Structura tezei

Lucrarea de doctorat este structurată în șapte capitole.

Primul capitol cuprinde o introducere în domeniul simulatoarelor de zbor și a algoritmilor de indicare a mișcării pentru diferite platforme de mișcare, platforme robotice paralele și seriale, fiind evidențiate obiectivele și motivația acestui studiu.

Al doilea capitol prezintă stadiul actual al cercetării privind modelarea matematică a algoritmilor de mișcare specifici simulatoarelor de zbor ce au ca obiectiv transmiterea către pilot a senzațiilor de accelerație pe care aceștia le-ar avea într-un avion real.

Capitolul trei cuprinde modelarea matematică a sistemului de simulare dinamică pentru aplicații de tip pilot-in-the-loop. Acest capitol începe cu o introducere a platformei de simulator, fiind prezentate modulele simulatorului și structura funcțională a acestuia. În următoarea parte a capitolului sunt prezentate sistemele de referință și transformările de coordonate utilizate în cadrul lucrării prezente. Capitolul cuprinde și modelarea matematică a sistemului robotic serial, fiind analizată și simulată cinematica directă și cinematica inversă a sistemului robotic.

În capitolul patru lucrarea continuă să prezinte procedura de scalare a spațiului de lucru al simulatorului de zbor pornind de la metoda de tratare a singularităților și optimizarea locală, astfel concluziile capitolului prezintă limita minimă și maximă a articulațiilor platformei de mișcare pentru configurația de testare a algoritmului de indicare a mișcării.

Capitolul cinci cuprinde modelarea matematică a algoritmului de indicare a mișcării. Acest capitol începe cu o introducere specifică filtrului de constrângere optimal bazat pe LQR, fiind prezentată structura generală a algoritmului. Apoi se continuă cu prezentarea modulului de scalare și limitare din cadrul algoritmului care are rol de a păstra sistemul robotic serial de tip

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

ABB IRB 7600-500 în limitările și constrângerile sigure pe parcursul utilizării simulatorului de zbor pentru testarea algoritmului. În următorul subcapitol este prezentat succint modelul matematic al sistemului vestibular care este un modul esențial din algoritmul de indicare a mișcării. Capitolul continuă cu prezentarea procedurii de implementare a algoritmului de indicare a mișcării bazat pe filtru de constrângere optimal. Capitolul se încheie cu modelul optimizat al algoritmului fiind utilizați algoritmi evolutivi de tip algoritmi genetici.

Capitolul șase prezintă scenariile de zbor pentru care a fost testat algoritmul de indicare a mișcării și rezultate specifice pentru scenariile analizate.

Teza se încheie cu un capitol de concluzii generale, contribuții proprii și perspective de dezvoltare a activității de cercetare în domeniul algoritmilor de indicare a mișcării specifici simulatoarelor de zbor. Bibliografia studiată și utilizată în realizarea activității de cercetare din teza de doctorat este, de asemenea, prezentată.

## **2. Stadiul actual al cercetării privind modelarea matematică a algoritmilor de mișcare specifici simulatoarelor de zbor**

Capitolul 2 prezintă o cercetare asupra stadiului actual a problematicii abordate privind dezvoltarea algoritmilor de indicare a mișcării. În cadrul lucrării de doctorat se abordează problematica soluționării modului de transmitere a indicilor de mișcare în cadrul simulatorului de zbor utilizând algoritmi de indicare a mișcării, denumiți și algoritmi filtru de constrângere.

### **2.1. Introducere în problematica algoritmilor de indicare a mișcării pentru simulatoare de zbor**

Filtrul de constrângere este un algoritm care are ca obiectiv să transmită pilotului senzațiile de accelerație pe care le-ar avea într-un avion real, pe o platformă de mișcare care are un spațiu de lucru considerabil restrâns ([13], [14], [15]). Acesta primește date de intrare accelerațiile liniare și vitezele unghiulare ale avionului simulat și furnizează ca date de ieșire traiectoria care trebuie urmărită de către efectorul final al platformei de mișcare în sistem de coordonate carteziane. Acest algoritm constă în combinarea a trei canale (rotațional, translațional, coordonare a înclinării) care sunt compuse fiecare din filtre trece-jos sau trece-sus, în funcție de comportamentul pe care simulatorul de zbor trebuie să îl reprezinte.

Cele mai importante teorii din acest domeniu sunt filtrele de tip: optimal ([16], [17], [18], [19], [20]), adaptiv ([15], [19], [21], [22], [23], [24]) și robust ([16], [18], [19], [25], [26], [27]).

### **2.2. Algoritm filtru de constrângere clasic**

Algoritmul filtru de constrângere clasic este soluția de bază care a fost utilizată pe scară largă în diferite tipuri de simulatoare datorită simplității și ușurinței de ajustare ([28], [29], [30], [31]). Alte avantaje ale algoritmului filtru de constrângere clasic includ: timp de procesare scurt și performanță stabilă. Pe baza studiilor realizate ([32], [33], [34]), ([35], [36], [13]) s-a concluzionat că un algoritm filtru de constrângere clasic și multe neajunsuri. Este inflexibil, deoarece necesită un proces de reglare care se concentrează pe situația cea mai nefavorabilă. Acest lucru duce la o mișcare conservatoare și la o utilizare deficitară a spațiului de lucru. Ca

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

urmare, algoritmul nu este potrivit pentru toate circumstanțele. Principalul dezavantaj al algoritmului filtrului de constrângere clasic este lipsa de luare în considerare a percepției umane, deoarece sistemul vestibular este ignorat în structura sa.

### 2.3. Algoritm filtru de constrângere control optimal

Algoritm filtru de constrângere control optimal se bazează pe senzația de mișcare umană și ia în considerare sistemul vestibular ([37], [38]). În esență, metoda integrează un model matematic al sistemului vestibular uman pentru a minimiza eroarea de senzație între pilotul de pe simulator și cel dintr-o cabină de avion reală. Metoda utilizează tehnici de control optimal bazate pe regulator liniar pătratic pentru dezvoltarea filtrelor de ordin superior înainte de aplicarea în timp real și a concepe o funcție de cost care depinde atât de eroarea de senzație dintre piloții aeronavelor reale și simulate, cât și de mișcarea platformei.

În dezvoltarea unui filtru de constrângere control optimal bazat pe metoda LQR, problema cheie este de a găsi o matrice de constrângere adecvată a funcțiilor de transfer liniar,  $W(s)$ . Această matrice pune în relație intrările simulatorului și ale mișcării vehiculului, în încercarea de a minimiza funcția de cost prin limitarea atât a erorii de senzație dintre simulator și pilotul aeronavei, cât și a mișcării platformei. În Figura 2.1 este ilustrată o schemă a filtrului de constrângere control optimal bazat pe LQR.

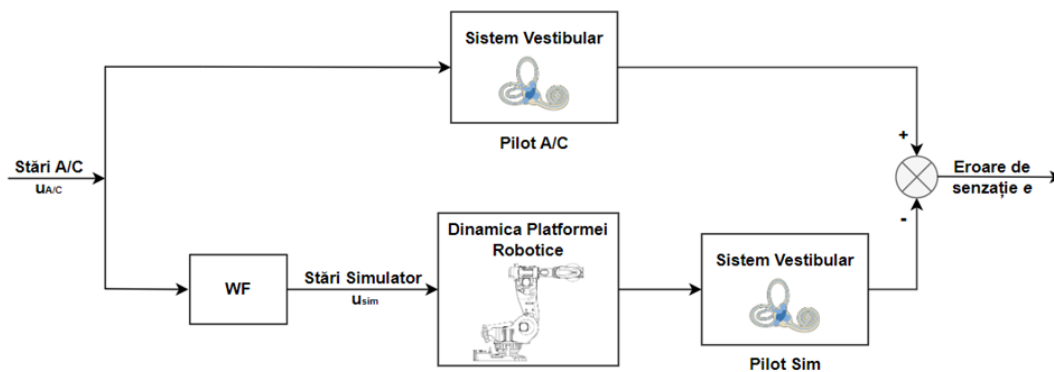


Figura 2.1 Structură algoritm control optimal bazat pe LQR, adaptare după [39]

### 2.4. Centralizarea caracteristicilor algoritmilor de indicare a mișcării

Pe baza studiului privind stadiul actual al cercetării în literatura de specialitate pentru proiectarea, dezvoltarea, implementarea și testarea unui algoritm de indicare a mișcării pentru simulatoare de zbor au fost identificate patru asemenea tipuri de algoritmi bazați pe:

- Filtre de constrângere clasice;
- Filtre de constrângere adaptive;
- Filtre de constrângere control optimal;
- Filtre de constrângere control optimal robust.

Principalele avantaje ale filtrului de constrângere clasic sunt [40]:

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

- simplu din punct de vedere matematic și computațional și, prin urmare, ieftin din punct de vedere computațional;
- relativ ușor de proiectat, dar este necesară o experiență în modificarea modelului bazat pe informațiile primite de la piloți în timpul testării experimentale a algoritmului în simulator.

Printre dezavantajele algoritmului filtru de constrângere clasic, pot fi menționate următoarele:

- utilizarea în principal a elementelor liniare, astfel încât nu se exploatează pe deplin capacitățile simulatorului și nici nu ia în considerare caracteristicile neliniare ale percepției mișcării umane (corpul uman este sensibil atât la forța specifică liniară, cât și la viteza unghiulară [16], experiență ce poate fi simulată prin implementarea unui model de sistem vestibular);
- trebuie să fie proiectat pentru cele mai critice stări, deoarece are parametri ficși care pot produce o mișcare minimă în cazul unor manevre mai blânde, limitare ce poate fi soluționată prin implementări mai elaborate ale filtrului de constrângere.

Filtrul de constrângere adaptiv (AWF) a fost propus de Parrish & colaboratorii săi [15] și Reid & colaboratorii [41]. Acesta are amplitudini de filtrare adaptive care variază pentru a minimiza o funcție de cost care penalizează eroarea de mișcare, magnitudinea mișcării și modificarea parametrilor adaptivi față de valorile lor inițiale [42]. Este important de subliniat faptul că doar amplitudinea filtrului trece-sus este adaptată, în timp ce amplitudinea filtrului trece-jos rămâne fixă. Reglarea unui astfel de filtru de constrângere se face prin alegerea ponderilor de cost și a amplitudinilor filtrului adaptiv.

Principalele avantaje ale acestei metode sunt următoarele [43]:

- generează indici de mișcare falși reduși în comparație cu filtrul de constrângere clasic;
- prezintă aceleași dificultăți de alegere a frecvențelor de tăiere, a valorilor amortizării și a amplitudinii inițiale ca și filtrul de constrângere clasic, dar cu toate acestea, alegerea ponderilor de cost este mai intuitivă pentru un utilizator cu puțină experiență, deoarece acestea sunt direct legate de erorile de accelerație și de viteză unghiulară, în loc să fie legate de frecvențe de tăiere care nu se referă în mod explicit la nicio eroare de mișcare;
- oferă indici de mișcare mai realiști datorită caracteristicilor adaptive atunci când simulatorul se află aproape de poziția neutră și reduce fidelitatea mișcărilor numai atunci când simulatorul se apropie de limitele sale fizice, oferind o mai bună utilizare a capacităților platformei de mișcare;
- poate sau nu să utilizeze modele vestibulare, precum și funcții non-quadratice care pot fi introduse pentru a varia penalizările [40], adică funcția de cost care trebuie minimizată este flexibilă.

Ca principale dezavantaje ale algoritmului filtru de constrângere adaptiv, pot fi considerate următoarele:

- din punct de vedere computațional comparativ cu filtrul de constrângere clasic este mai greoi;
- frecvențele de tăiere, amortizarea și valorile inițiale ale amplitudinii filtrului sunt reglate pe baza accelerațiilor din cel mai critic caz;
- minimizează eroarea de mișcare (accelerații carteziene și viteze unghiulare) între avion și platformă.



## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Pentru a considera eroarea de percepție ca fiind o variabilă care trebuie minimizată, Sivan și colaboratorii săi [14] au propus un filtru de constrângere bazat pe control optimal (OWF). Filtrul de constrângere rezultat a fost obținut dintr-un regulator pătratic liniar (LQR) care minimizează o funcție de cost care ia în considerare eroarea de percepție, deplasarea liniară față de poziția inițială și deplasările unghiulare și viteza, precum și comenzile de mișcare ale platformei. Acest filtru are următoarele caracteristici avantaje:

- minimizează eroarea de senzație a pilotului în loc de eroarea de mișcare propriu-zisă prin includerea unui model vestibular;
- ia luat în considerare coeficientul de corelație, considerat în schema filtrului de constrângere optimal [16], ca fiind un criteriu de urmărire a formei care ajută la generarea de semnale care pot urmări mai precis semnalele de referință;
- ușurință în reglare de către un utilizator cu puțină experiență, deoarece se reglează prin ajustarea ponderilor funcției de cost legate de variabile mai intuitive.

Dezavantajele acestui filtru sunt următoarele:

- schema de control optimal produce filtre cu parametru fix, similar cu schema clasică, care nu exploatează pe deplin capacitățile de mișcare ale platformei de mișcare și trebuie ajustate pentru cele mai critice manevre;
- limitarea vitezei de înclinare nu este inclusă în acest algoritm deoarece a avut efecte negative asupra comportamentului algoritmului [41];
- nu se iau în considerare în mod explicit constrângerile sistemului [16].

Tabelul 2.1 prezintă un rezumat al caracteristicilor schemelor filtru constrângere existente bazat pe cercetarea realizată în prezentul capitol, fiind luate în considerare avantajele și dezavantajele fiecărui filtru raportate la timpul computațional, dificultatea de reglare a algoritmului, includerea modelului de sistem vestibular uman și utilizarea crescută a spațiului de lucru al platformei de mișcare. Totodată, în tabel este prezentat și tipul de sistem robotic pentru care a fost proiectat, implementat și testat algoritmul de indicare al mișcării până în prezent, și anume Sisteme Robotice Paralele (notate cu *SRP*), respectiv Sisteme Robotice Seriale (notate cu *SRS*). Culoarea verde reprezintă un avantaj, iar culoarea gri un dezavantaj.

Tabel 2.1 Caracteristici algoritmi de indicare a mișcării

	CWF	AWF	OWF	RWF	CWF/Q
<b>Timp computațional redus</b>					
<b>Reglare ușoară algoritm</b>					
<b>Eroare de percepție umană</b>					
<b>Calitate uniformă - spațiu de lucru</b>					
<b>Implementare sistem robotic SRP / SRS</b>	SRP / SRS	SRP	SRP	SRP	SRP / SRS

Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

### 3. Modelare matematică a unui sistem de simulare dinamică pentru aplicații de tip pilot-in-the-loop

#### 3.1. Introducere – prezentare generală a sistemului de simulare dinamică

Platforma dedicată sistemului de simulare dinamică pentru aplicații de tip pilot-in-the-loop (Figura 3.1) are ca scop realizarea unei interconectări complexe de sisteme ce modelează simularea zborului, prin integrarea modelului de simulator de zbor bazat pe o platformă robotică serială. Obiectivul este crearea unui simulator de zbor cu 6 grade de libertate, în buclă închisă prin exploatarea brațului robotic ca platformă de mișcare.

Întreg sistemul propus constă într-un braț robotic manipulator, robot industrial de tip ABB IRB 7600-500/2.55, cu șase axe independente, la extremitatea căruia este montată o celulă de tip cabină de aeronavă. Ansamblul efectuează o mișcare de-a lungul unei șine de rulare, GUDEL TMF-4 V2, cu o cursă de 20 metri. Celula montată la extremitatea brațului robotic asigură redundanța cinematică necesară pentru a face față constrângerilor de acționare a robotului. Scopul celulei este de a crea un spațiu de tip cockpit, specific domeniului aerospațial, fiind dotat cu echipamente de avionică specifice, sistem vizual bazat pe realitate virtuală și sistem audio integrat pentru a oferi pilotului cât mai multe tipuri de indici care să imite ceea ce se întâmplă într-o cabină reală de pilotaj.

Totodată, este utilizată o interfață de conectare cabină-robot, utilă pentru a crea posibilitatea de transmitere a datelor din cabină către partea de la sol (camera de monitorizare și control), și invers, astfel încât să se creeze o comunicație justificată bidirecțională între cele două medii. Configurația propusă are ca scop principal exploatarea platformei de simulator de zbor ca mediu de simulare versatil pentru o gamă largă de scenarii de zbor, crescând fidelitatea mediului de simulare pentru analiza stabilității dinamice pentru noi configurații de aeronave, cum ar fi arhitectură de tip hibrid-electric.



Figura 3.1 Sistem de simulare dinamică pentru aplicații de tip pilot-in-the-loop, Credit: INCAS

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Implementarea acestei platforme de simulator cu mișcare urmărește realizarea studiilor de testare a percepției mișcării, a interacțiunii om-mașină, a psiho-fiziologiei umane și de asemenea, a evaluării comportamentului de control al pilotului în misiuni experimentale de bază.

Structura de simulare a avionului pentru sistemul de mișcare bazat pe platforma robotică este prezentată în Figura 3.2. Intrările date de către operator (pilot) sunt transmise către modelul dinamic al avionului, generând vectorul de stare al avionului. Trecând vectorul de stare al avionului prin algoritmul de indicare a mișcării se produc indicii de mișcare doriți și stările platformei robotice care oferă mișcarea simulatorului. Stările dorite ale platformei robotice sunt apoi transformate în spațiul de lucru al axelor, generând comenzi către cele șase axe ale robotului.

Comenzile de mișcare ale axelor servesc ca intrări în dinamica platformei robotice, rezultând mișcarea reală a simulatorului.

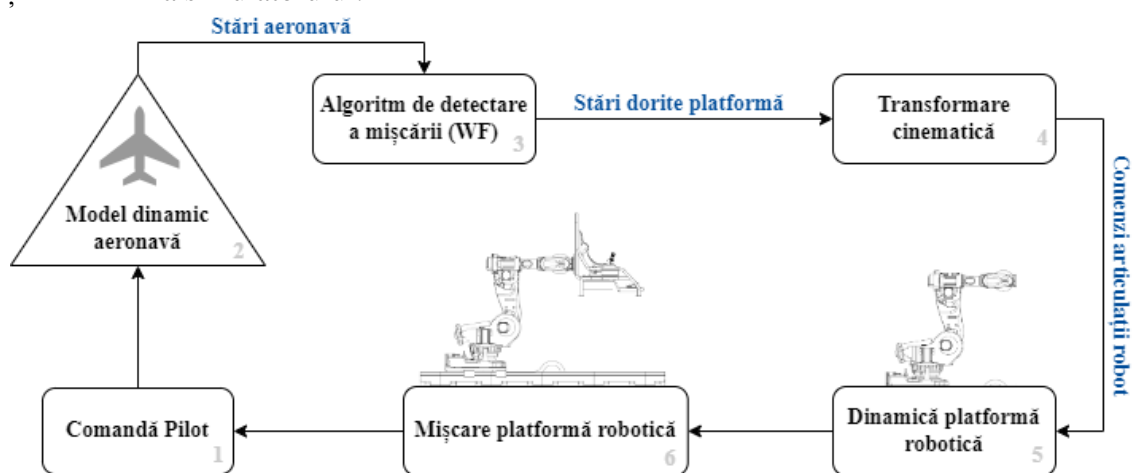


Figura 3.2 Schemă structură de simulare avion

Semnalele (intrările) operatorului (pilotului) sunt transferate în modelul de simulare dinamică executat pe o platformă în timp real. Semnalele date de operator (pilot) pot proveni de la diferite echipamente de comandă specifice, cum ar fi, poziția flapsului, acționare paloniere, acționare manetă de gaze, etc. Ieșirile din modelul dinamic de aeronavă se măsoară în accelerații translaționale, unghiuri de rotație și viteze unghiulare; ieșiri ce sunt necesare pentru reproducerea feedback-ului mișcării către simulator. Scopul este acela de a induce o mișcare apropiată de realitate în cabina de simulare. Este evident că, din cauza limitărilor spațiului de lucru al simulatorului, mișcarea nu poate fi identică cu cea a aeronavei și trebuie comprimată. Această caracteristică este furnizată de algoritmi de indicare a mișcării. După determinarea poziției și orientării cabinei, comanda individuală introdusă de robotul cu șase grade de libertate se calculează din transformarea cinematicii inverse. Intrările de comandă obținute trec prin filtrul de prevenire a coliziunii și sunt trimise robotului. Cu această etapă, bucla de reacție a mișcării pentru pilotul de teste se închide.

Simulatorul de zbor bazat pe sistemul robotic serial cu șase axe cuprinde o dezvoltare a componentelor simulatorului ca un sistem deschis compus din module modificabile și scalabile. Cele șase grade de libertate permit simularea mișcărilor de rotație ale avionului (ruliu, tangaj și girație), precum și deplasările în coordonate (X, Y, Z) ale sistemului cartezian.

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Principalele componente ale modelului de aeronavă dedicată sistemului de simulator de zbor sunt prezentate în Figura 3.3.

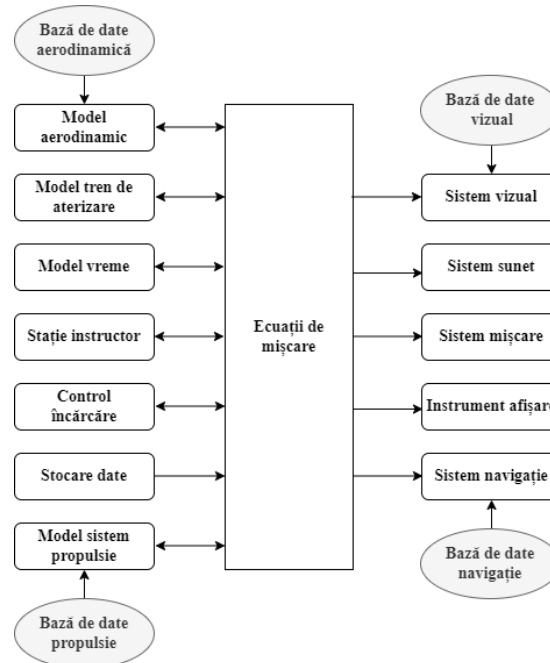


Figura 3.3. Componente model simulator de zbor

Sistemul de simulare dinamică pentru aplicații de tip pilot-in-the-loop bazat pe platformă cu mișcare acoperă simularea în trei zone:

- Simularea dinamicii aeronavelor (model de zbor aerodinamic și motor, sisteme de aeronave, manipulare la sol, comenzi de zbor, configurație cabină).
- Simularea mediului (mediul ATC, mediul înconjurător, teren și aerodrom, navigație).
- Simularea mișcării (indici de mișcare, indici vizuali, indici audio).

### 3.2. Sisteme de coordonate

În definierea și implementarea algoritmului de indicare a mișcării se utilizează o serie de sisteme de referință. Aceste sisteme de referințe sunt definite în prezentul capitol și sunt prezentate în Figura 3.4.

#### 3.2.1. Sistem de referință avion

Sistemul de referință avion  $SR_{A/C}$  are originea în centrul de greutate al aeronavei, notat  $CG$ . Sistemul de referință  $SR_{A/C}$  are o orientare pentru  $X_{CG}$ ,  $Y_{CG}$  și  $Z_{CG}$ , care are o orientare paralelă cu sistemul de referință simulator  $SR_{sim}$  și sistemul de referință pilot-avion  $SR_{AP}$ .

#### 3.2.2. Sistem de referință simulator

Sistemul de referință simulator  $SR_{sim}$  își are originea în centrul articulației 6 a robotului de care este atașat simulatorul, mai precis în centrul contra-pieseii de prindere a celulei de simulator de robot.  $X_{sim}$  este îndreptat în față și  $Z_{sim}$  este îndreptat în sus în raport cu cabina de pilotaj a simulatorului, iar  $Y_{sim}$  este îndreptat spre partea dreaptă a pilotului. Planul  $X - Y$  este paralel cu podeaua cabinei de pilotaj.

### 3.2.3. Sistem de referință pilot-avion

Sistemul de referință pilot-avion  $SR_{AP}$  își are originea în aceeași poziție relativă a cabinei de pilotaj ca și sistemul de referință al simulatorului  $SR_{sim}$ .  $SR_{AP}$  are aceeași orientare pentru  $X_{AP}$ ,  $Y_{AP}$  și  $Z_{AP}$  în raport cu cabina de pilotaj ca și sistemul de referință al simulatorului  $SR_{sim}$ .

### 3.2.4. Sistem de referință pilot-simulator

Sistemul de referință pilot-simulator  $SR_{SP}$  își are originea în aceeași poziție relativă a cabinei de pilotaj ca și sistemul de referință al simulatorului  $SR_{sim}$ .  $SR_{SP}$  are orientarea pentru  $X_{SP}$  identică cu orientarea  $X_{sim}$ ,  $Y_{SP}$  este îndreptat spre partea dreaptă a pilotului, iar  $Z_{SP}$  este orientat în jos în raport cu cabina de pilotaj.

### 3.2.5. Sistem de referință inerțial

Sistemul de referință  $SR_I$  este fixat de Pământ, cu  $Z_I$  aliniat cu vectorul gravitațional  $g$ . Originea sa este situată în centrul bazei platformei de mișcare.  $X_I$  este îndreptat spre înainte, iar  $Y_I$  este îndreptat spre partea dreaptă în raport cu pilotul simulatorului. În Figura 3.4 sunt reprezentați vectorii care definesc poziția relativă a sistemelor de referință.  $R_{sim}$  definește locația  $SR_{SP}$  în raport cu  $SR_{sim}$ . Similar,  $R_{A/C}$  definește locația sistemului de referință  $SR_{AP}$  în raport cu  $SR_{A/C}$ .

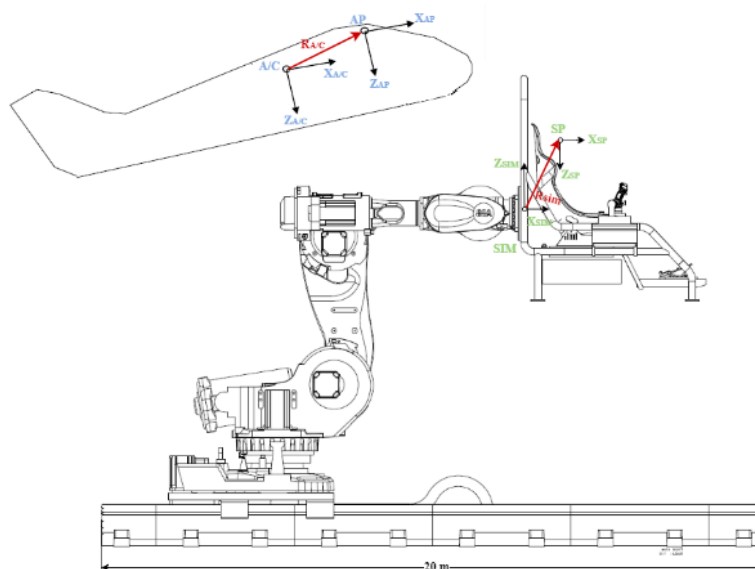


Figura 3.4 Poziționare sisteme de referință

## 3.3. Transformări de coordonate

Orientarea între sistemul de referință al avionului  $SR_{A/C}$  și sistemul de referință al simulatorului  $SR_{sim}$  se realizează prin utilizarea matricei cosinusurilor directe, printr-o rotație ZYX.

$$SR_{A/C} = DCM_{ZYX} \cdot SR_{sim} \quad 3.1$$

unde

$$DCM = \begin{bmatrix} c(\theta) \cdot c(\psi) & c(\theta) \cdot s(\psi) & -s(\theta) \\ s(\phi) \cdot s(\theta) \cdot c(\psi) - c(\phi) \cdot s(\psi) & s(\phi) \cdot s(\theta) \cdot s(\psi) + c(\phi) \cdot c(\psi) & s(\phi) \cdot c(\theta) \\ c(\phi) \cdot s(\theta) \cdot c(\psi) + s(\phi) \cdot s(\psi) & c(\phi) \cdot s(\theta) \cdot s(\psi) - s(\phi) \cdot c(\psi) & c(\phi) \cdot c(\theta) \end{bmatrix} \quad 3.2$$

### 3.4. Modelarea matematică a sistemului robotic serial

În prezentul subcapitol se prezintă modulul cinematicii inverse pentru sistemul robotic serial ce apare ca modul de legătură între modulul specific algoritmului de indicare a mișcării și modulul specific dinamicii platformei robotice în schema generală de funcționare a simulatorului de zbor.

Figura 3.5 prezintă structura cinematică a robotului ABB IRB 7600-500. În Figura 3.5a sunt prezentate actuatorii robotului plasate la articulații între elementele lanțului de acționare în serie, iar în Figura 3.5b sunt reprezentate sisteme de referință specifice fiecărei articulații ale robotului.

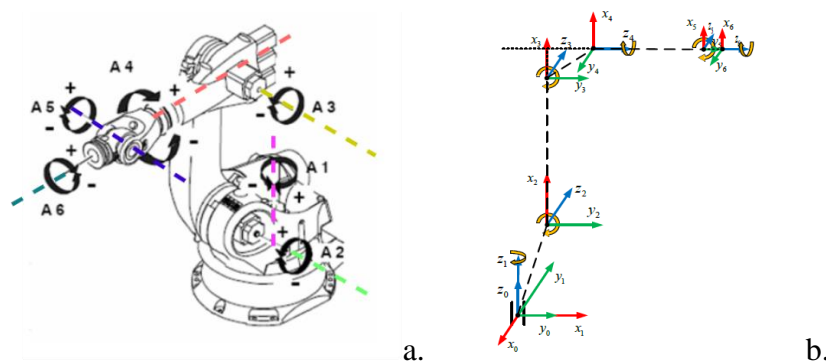


Figura 3.5 Structură generală robot serial IRB 7600-500 (a. Axe robot, b. Sisteme de referință axe robot)

Cinematica robotului este împărțită în cinematică directă și cinematică inversă.

Corelația dintre cinematica directă și cinematica inversă este prezentată în Figura 3.6 [44].

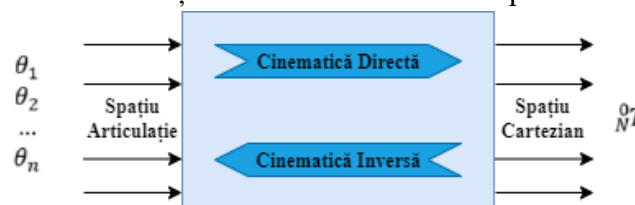


Figura 3.6 Descriere schematică a cinematicii directe și inverse a robotului serial

#### 3.4.1. Cinematica directă a robotului serial

Pentru studiul cinematicii directe a robotului serial se utilizează reprezentarea matematică Denavit-Hartenberg (D-H) [45], care arată că o transformare globală între două articulații necesită patru parametri care descriu cinematica robotului. Denavit-Hartenberg (D-H) oferă

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

modificarea comună între două articulații, luând în considerare lungimile elementelor și unghiurile articulațiilor.

Pentru modelarea robotului ABB IRB 7600-500/2.55 au fost utilizate următoarele două instrumente: MATLAB R2022b Robotics System Toolbox [46] și Robotics Vision Control Toolbox [47], ambele respectând terminologia Denavit și Hartenberg (D-H).

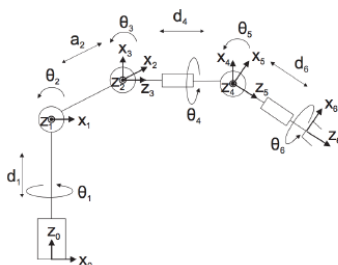


Figura 3.7 Diagramă robot IRB 7600-500

În conformitate cu sistemul de coordonate stabilit, parametrii corespunzători ai elementelor sunt definiți astfel:

- $a_{i-1}$  [mm]: distanța măsurată de la  $z_{i-1}$  la  $z_i$  de-a lungul lui  $x_{i-1}$ ;
- $\alpha_{i-1}$  [°]: unghiul de rotație de la  $z_{i-1}$  la  $z_i$  în jurul axei  $x_{i-1}$ ;
- $d_i$  [mm]: distanța măsurată de la  $x_{i-1}$  la  $x_i$  de-a lungul lui  $z_i$ ;
- $\theta_i$  [°]: unghiul de rotație de la  $x_{i-1}$  la  $x_i$  în jurul axei  $z_i$ .

Acești patru parametri descriu lanțurile cinematice ale robotului. Parametrii Denavit - Hartenberg (D-H) pentru robotul IRB 7600-500/2.55 sunt prezentați în Tabelul 3.1. În calcularea parametrilor D-H au fost utilizate date de intrare specifice sistemului robotic precum: dimensiune elemente (Figura 3.8), direcția de rotație/deplasare a articulațiilor (Figura 3.5a) și sistemul de referință de bază al robotului (Figura 3.5b).

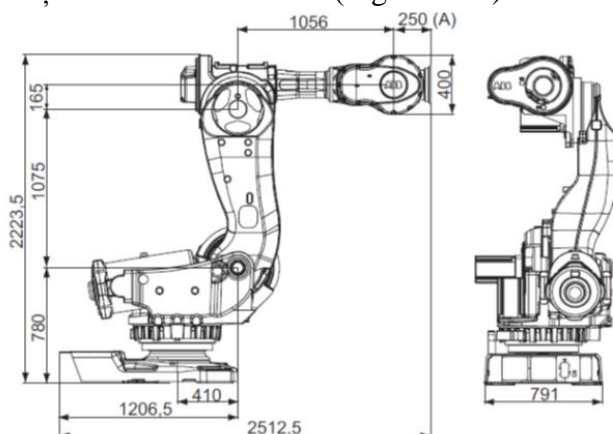


Figura 3.8 Schiță specificații dimensiuni robot ABB IRB 7600-500/2.55 [48]

Tabel 3.1 Parametrii D-H și limitele articulațiilor sistemului robotic ABB IRB 7600-500/2.55

Element de legătură $i$	$a_{i-1}$ [mm]	$\alpha_{i-1}$ [°/rad]	$d_i$ [mm]	$\theta_i$ [°/rad]	Articulație limită min/max [°/rad]	Articulații robot IRB 7600-500
1	0.41	-90	0.78	0	- 180 / +180	$q_1$

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

		$-\pi/2$			$-\pi/\pi$	
<b>2</b>	1.075	0	0	-90	- 60 / +85	$q_2$
				$-\pi/2$	$-\pi/3/ \pi/3$	
<b>3</b>	0.165	-90	0	0	- 180 / +60	$q_3$
		$-\pi/2$			$-\pi / \pi/3$	
<b>4</b>	0	90	1.056	0	- 300 / +300	$q_4$
		$\pi/2$			$-5\pi/3 / 5\pi/3$	
<b>5</b>	0	-90	0	0	- 100 / +100	$q_5$
		$\pi/2$			$-5\pi/9 / 5\pi/9$	
<b>6</b>	0	0	0.25	180	- 360 / +360	$q_6$
				$\pi$	$-2\pi / 2\pi$	

Cinematica directă a robotului calculează poziția și atitudinea dispozitivului de acționare finală – efectorul final ( $EE$ ), luând în considerare valorile celor șase articulații. Formula generală de transformare a elementului  ${}^{i-1}_i T$  în cinematica brațului robotic este:

$${}^{i-1}_i T = T_x(\alpha_{i-1})T_x(a_{i-1})T_z(\theta_i)T_z(d_i) \quad 3.3$$

unde  ${}^{i-1}_i T$  este o matrice de transformare omogenă,  $\alpha_{i-1}$  și  $a_{i-1}$  sunt translația și rotația în jurul axei  $x_{i-1}$ , respectiv  $d_i$  și  $\theta_i$  sunt translația și rotația în jurul axei  $z_{i-1}$ .

Cinematica directă a manipulatorului este exprimată prin:

$${}^0T(q_1){}_2^1T(q_2){}_3^2T(q_3){}_4^3T(q_4){}_5^4T(q_5){}_6^5T(q_6) = T_f \quad 3.4$$

### 3.4.2. Cinematica inversă a robotului serial

Soluția cinematicii inverse a robotului - înseamnă că valoarea variabilelor articulațiilor robotului este determinată în funcție de poziția dată a efectorului final și de atitudinea manipulatorului [49]. Metoda de rezolvare a transformării inverse constă în multiplicarea inversă a unei anumite transformări pe ambele părți ale ecuației cinematicii robotului în același timp. Formula specifică este exprimată după cum urmează:

$${}^0T^{-1}(q_i){}_6^0T = {}_{i+1}^i T(q_{i+1}){}_{i+2}^{i+1} T(q_{i+2}){}_{i+3}^{i+2} T(q_{i+3}) \dots {}_6^5 T(q_6) \quad 3.5$$

Deoarece poziția elementului de capăt  ${}_6^0 T$  este cunoscută, se poate obține produsul dintre inversul fiecărei transformări din partea stângă a ecuației și transformarea din partea dreaptă a ecuației.

$${}^0T = {}_1^0 T(q_1){}_2^1 T(q_2){}_3^2 T(q_3){}_4^3 T(q_4){}_5^4 T(q_5){}_6^5 T(q_6) \quad 3.6$$

Pentru a găsi mai întâi  $q_1$  în funcție de elementele cunoscute, se calculează inversarea transformării legăturii care sunt pre-multiplicate după cum urmează  ${}_1^0 T(q_1)$ :

$${}^0T^{-1}(q_1){}_6^0T = {}^0T^{-1}(q_1){}_1^0 T(q_1){}_2^1 T(q_2){}_3^2 T(q_3){}_4^3 T(q_4){}_5^4 T(q_5){}_6^5 T(q_6) \quad 3.7$$

unde  ${}^0T^{-1}(q_1){}_1^0 T(q_1) = I$ , iar  $I$  este matricea identitate.



## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

În final vor exista 12 seturi simultane de ecuații neliniare care trebuie rezolvate. Cele 12 elemente ale matricei neliniare din partea dreaptă sunt fie zero, fie constante, fie funcții de la  $q_2$  la  $q_6$ .

În soluționarea problemei de cinematică inversă a robotului serial cu 6 DoF [44] s-a utilizat algoritmul de optimizare Broyden–Fletcher–Goldfarb–Shanno (BFGS) și s-a simulat soluția cinematicii inverse folosind două medii de simulare diferite: MATLAB R2022b și Robot Studio [50]. Primul proces de simulare utilizează Matlab Robotics System Toolbox [46] pentru a verifica poziția efectorului final și pentru a determina unghiurile articulațiilor pentru fiecare punct stabilit în traiectoria propusă. Al doilea proces de simulare utilizează programul RAPID [51] din Robot Studio [50], pentru a valida rezultatele înregistrate în prima simulare, utilizând același algoritm. Traectoria și mișcarea robotului, precum și urmărirea unghiului articulației sunt descrise în limbajul de programare RAPID care scrie datele pentru analiză. Determinarea poziției și atitudinii efectorului final a fost realizată cu ajutorul unor funcții speciale din programul Robot Studio (CalcRobT și CalcJointT). Funcțiile Robot Studio găsesc traiectoria definită prin cinci puncte diferite, iar poziția efectorului final pentru aceste cinci puncte este prezentată în tabelul 3.3. Ultimele trei coloane din tabelul 3.3 evidențiază abaterile exprimate în [mm].

Tabel 3.2 ABB IRB 7600-500 - Poziție de capăt pentru o traiectorie specificată

Punct	Puncte traiectorie [mm]			Poziție efector final [mm]			Abatere [mm]		
	x	y	z	x	y	z	$\Delta x$	$\Delta y$	$\Delta z$
1	1900.0	100.0	1133.3	1899.85	100.227	1133.52	0.1500	-0.2270	-0.2200
2	1838.4	332.4	1128.3	1838.60	331.704	1128.32	-0.2000	0.6960	-0.0200
3	1677.8	502.8	1113.2	1678.12	502.442	1113.18	-0.3200	0.3580	-0.0200
4	1469.9	585.0	1088.7	1470.18	584.900	1088.67	-0.2800	0.1000	0.0300
5	1258.9	587.8	1055.3	1258.90	587.800	1055.30	0	0	0

Pentru fiecare dintre punctele 1, 3 și 5 din traiectorie, valorile unghiurilor articulațiilor (unghi articulație C) au fost calculate prin cinematică inversă și comparate cu cele obținute prin simularea mișcării robotului de serie ABB IRB 7600-500 folosind Robot Studio, prin programul RAPID (unghi articulație R). Din rezultatele evidențiate în tabelul de mai jos se observă că există diferențe mici între valorile unghiurilor articulațiilor.

Tabel 3.3 Valorile unghiurilor articulațiilor manipulatorului

Punct 1			Punct 3			Punct 5		
Articulație unghi C	Articulație unghi R	$\Delta$ _Articulație unghi	Articulație unghi C	Articulație unghi R	$\Delta$ _Articulație unghi	Articulație unghi C	Articulație unghi R	$\Delta$ _Articulație unghi
3.4759	3.4682	0.0077	19.3838	19.3997	-0.0159	30.2200	30.2257	-0.0057
21.4093	21.4218	-0.0125	6.0040	15.9947	0.0093	2.5363	2.5247	0.0116
24.7475	24.7419	0.0056	33.8021	33.8135	-0.0114	54.6952	54.7075	-0.0123
4.8087	4.8028	0.0060	24.7384	24.7495	-0.0111	34.7130	34.7169	-0.0039
-46.2650	-46.2643	-0.0007	-52.4983	-52.5038	0.0055	-62.1140	-62.1179	0.0039
-3.3290	-3.3244	-0.0046	-15.6684	-15.6748	0.0064	-17.9533	-17.9535	0.0002

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Rezultatele grafice ale traiectoriei predefinite în mediile MATLAB și Robot Studio pentru simulările cinematice inverse prezentate în lucrarea [44] sunt evidențiate în Figura 3.9.

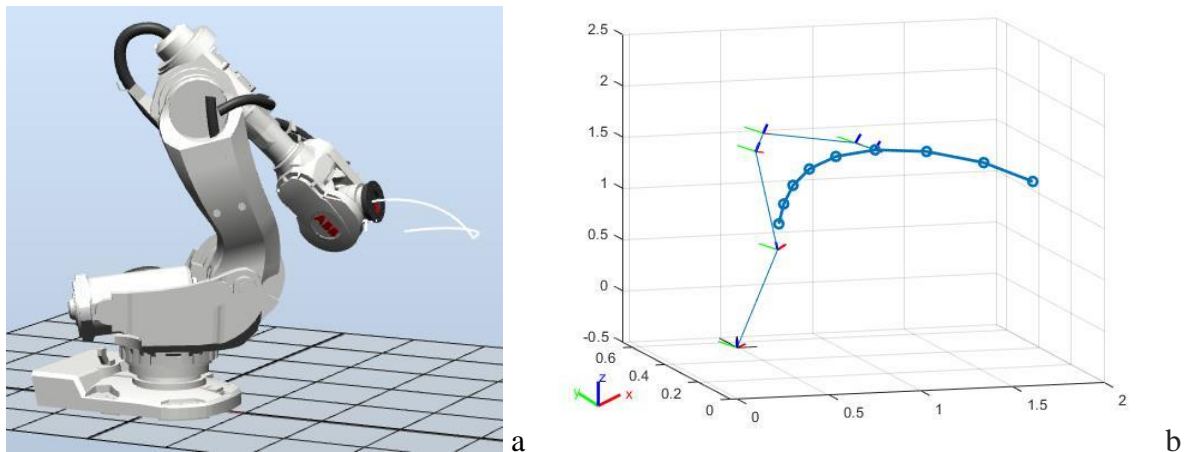


Figura 3.9 Traiectorie simulată robot IRB7600-500 (a. RobotStudio, b. MATLAB)

În acest subcapitol a fost propusă o soluție analitică a cinematicii directe și inverse a robotului de serie ABB IRB 7600-500. A fost testată pentru o utilizare eficientă precizia modelelor în mediul de simulare. Pentru a obține soluția, a fost luată în considerare o abordare geometrică prin utilizarea relațiilor geometrice între diferitele elemente ale manipulatorului. Modelul cinematicii inverse a fost simulat în două medii diferite și s-a observat că efectorul final s-a deplasat de-a lungul unei traiectorii predefinite care a confirmat eficacitatea modelului. Scopul acestei analize este de a valida modelul de cinematică inversă dezvoltat în mediul de simulare MATLAB care este parte din programul dezvoltat în același mediu de simulare ce servește ca platformă de testare offline a algoritmului de indicare a mișcării. De asemenea, a fost dezvoltată o reprezentare grafică a poziției articulației în funcție de timp. Acest studiu a fost realizat pentru a construi o bază pentru îmbunătățirea problemelor de robotică în domeniul aerospațial, când sunt utilizate ca platforme de mișcare pentru simulatoare de zbor.

### 4. Scalarea spațiului de lucru pentru sistemul de simulare dinamică

Pentru a oferi un spațiu de lucru sigur și stabil pentru simulatorul cu șase grade de libertate bazat pe platforma robotică serială IRB 7600-500 a fost necesară realizarea procedurii de scalare a spațiului de lucru prin definirea unor limitări hardware și software.

Intervalele de mișcare ale robotului au fost strict limitate din motive de siguranță astfel încât pilotul să opereze în condiții de siguranță simulatorul, evitându-se coliziunea cu alte obiecte aflate în proximitatea zonei de lucru. Aceste modificări au condus la alte constrângeri, micșorându-se astfel zona de lucru a robotului. Comparând zona de lucru a robotului IRB 7600-500/2.55 cu zona de lucru al robotului cu cabina de simulator cuplată se observă că spațiul de lucru al simulatorului este mai restrâns datorită dimensiunilor celulei de simulator.

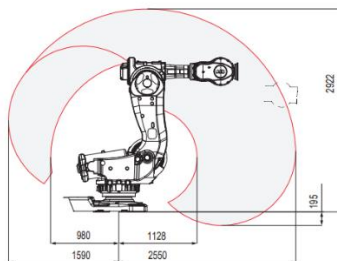


Figura 4.1 Spațiu de lucru robot IRB 7600 -500 [48]

În funcție de aplicația în care este utilizat simulatorul de zbor se definesc diferite constrângeri privind spațiul de lucru. De menționat este faptul că între unghiurile articulațiilor robotului există o dependență, astfel limitarea unei articulații are efect asupra celorlalte articulații. Prin urmare, o maximizare independentă a fiecărui interval de unghi al articulațiilor este imposibilă. În consecință, a fost formulată următoarea problemă de optimizare:

$$\max \vec{\Omega} \cdot \vec{f}_c(\vec{q}), \vec{q} \in R^6 \quad 4.1$$

cu

$$\vec{f}_c(\vec{q}) = \{\Delta X, \Delta Y, \Delta Z, \Delta \phi, \Delta \theta, \Delta \psi\}^T \quad 4.2$$

unde  $X, Y, Z$  sunt pozițiile carteziene,  $\phi, \theta, \psi$  unghiurile de rotație (ruliu, tangaj, girație), iar  $\vec{\Omega} \in R^6$  vectorul factorului de ponderare care reprezintă constrângerile de limitare a articulațiilor pentru a controla mișcarea robotului în limitele unghiurilor admise ale articulațiilor.

#### 4.1.Limitări hardware ale unghiurilor articulațiilor

Intervalele unghiurilor articulațiilor ale robotului sunt specificate de către producător în fișa tehnică. Aceste limitări hardware sunt însoțite și de limitări software impuse de producător și sunt declanșate (activate) înainte de a ajunge în limita hardware, ca măsură de siguranță. Intervalul configurației articulațiilor robotului este evidențiat prin:

$$\vec{q} \in [\vec{q}_{min}, \vec{q}_{max}] \quad 4.3$$

Tabel 4.1 Definiere limite robot IRB 7600

Robot IRB 7600-500/2.55	Axe	Limită min articulație $\vec{q}_{min}$	Limită max articulație $\vec{q}_{max}$	Viteză limită	Accelerații limită
A	$q_1$	- 180 [°]	+ 180 [°]	75 [°/s]	-
B	$q_2$	-60 [°]	+85 [°]	50 [°/s]	-
C	$q_3$	-180 [°]	+60 [°]	55 [°/s]	-
D	$q_4$	- 300 [°]	+ 300 [°]	100 [°/s]	-
E	$q_5$	- 100 [°]	+100 [°]	100 [°/s]	-
F	$q_6$	- 360 [°]	+ 360 [°]	160 [°/s]	-

<b>Șina de rulare</b>	$q_7$	-10 [mm]	20010 [mm]	90 [m/min]	86 [m/s <sup>2</sup> ]
-----------------------	-------	----------	------------	---------------	------------------------

#### 4.2. Eliminarea coliziunilor

Modul de proiectare al celulei de simulator are un efect semnificativ asupra definirii intervalelor de mișcare ale articulațiilor. Au fost luați în considerare următorii parametri în etapa de definire a condiției de eliminare a coliziunilor: dimensiunea cabinei, poziția flanșei, unghiul flanșei și dimensiunea subiectului (pilot). Siguranța sistemului trebuie să fie garantată pe întreaga plajă de mișcări ale robotului, astfel trebuie să se asigure că nu va exista nicio coliziune în contextul diferitelor combinații de mișcări. Următoarele obiecte au fost luate în considerare: podeaua, șina Gudel TMF-4, robotul IRB 4600, controlerul IRC5 a robotului IRB 4600, controlerul IRC5 a robotului IRB 7600 și peretele aflat în proximitatea șinei.

$$V(\text{obiect}_i, \vec{q}) \cap V(\text{obiect}_k, \vec{q}) = \emptyset \quad 4.4$$

unde  $\text{obiect}_i = \{\text{cabina simulator, robot IRB 7600, pilot, ...}\}$ ,

iar  $\text{obiect}_k = \{\text{robot IRB 4600, șina Gudel TMF - 4, controler IRC5, perete, ...}\}$ .

#### 4.3. Limitare software ale unghiurilor articulațiilor

În funcție de tipul de simulare, este necesară aplicarea diferitelor tipuri de constrângeri și limitări ale spațiului de lucru. Astfel în urma simulărilor offline au fost stabilite mai multe limitări software, precum instalarea modului SafeMove, implementări software în controlerul IRC5 și limitări ale axei  $q_5$  pentru a evita eroarea de singularitate.

Formularea problemei de generare a unei traiectorii bazate pe optimizare se bazează pe metoda de tratare a singularităților, fiind modificată pentru a satisface cerințele algoritmului de planificare a traiectoriei.

Fie

$$x_e = f(q) \quad 4.5$$

cinematica directă a brațului robotului, unde poziția efectorului este notată cu  $x_e$ , pozițiile articulațiilor sunt notate cu  $q$ .

Cinematica inversă este reprezentată astfel:

$$q = f^{-1}(x_e) \quad 4.6$$

Acest lucru implică faptul că există o multitudine de traiectorii comune  $q(t)$  care conduc la aceeași traiectorie a efectorului  $x_e(t)$ . O abordare pentru a calcula cinematica inversă dorită pentru roboții seriali este minimizarea erorii dintre poziția de referință și cea a efectorului în prezența constrângerilor, cu pozițiile articulațiilor robotului ca variabile de optimizare [52].

$$\min \|x_{eref} - x_{eq}\| \quad 4.7$$

unde  $x_{eref}$  este poziția de referință calculată. După cum este ilustrat în Figura 4.2, traiectoria  $q^i = q(t_i)$  este calculată ca rezultat al unei optimizări în fiecare pas de eșantionare.

În apropierea singularităților, limitele dinamice ale sistemului și stabilitatea soluției sunt garantate de constrângeri, dar pentru cazurile de imposibilitate mecanică se introduce o eroare

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

de poziționare. Această abordare calculează mișcarea optimă la nivel local a sistemului robotizat pentru a îndeplini sarcinile date de criteriile de optimizare.

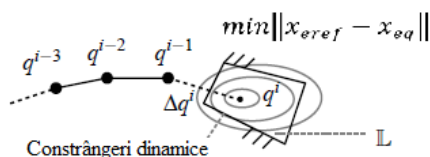


Figura 4.2 Optimizare locală

Utilizarea optimizării locale pentru a determina următorul unghi al articulației în fiecare pas de timp este prezentată în Figura 4.2. Problema de optimizare  $\min \|x_{eref} - x_{eq}\|$  se rezolvă considerând constrângeri dinamice ale robotului, definind astfel un spațiu de posibile soluții. Pentru a lua în considerare comportamentul dinamic al unui sistem în problema de optimizare, ecuațiile diferențiale sunt reformulate sub forma unor constrângeri ale optimizării.

Utilizarea unei configurații robotice seriale ca platformă de mișcare pentru simulatoare de zbor introduce unele probleme în ceea ce privește planificarea traiectoriei. În apropierea configurațiilor singulare, sunt necesare viteze unghiulare foarte mari ale articulațiilor pentru a menține traiectoria dorită. În cazul în care traiectoria planificată ajunge în apropierea unei singularități, limitele vitezelor de acționare conduc la erori de traiectorie și provoacă o oprire în caz de avarie de către comanda robotului.

Ținând cont de constrângerile specifice ale unghiurilor articulațiilor și de cerințele în raport cu spațiul de lucru, intervalele unghiurilor de articulație  $q_3$ ,  $q_4$  și  $q_5$  sunt specificate ca și condiții laterale pentru problema de optimizare. Aceste intervale de unghiuri ale articulațiilor au o influență puternică asupra spațiului de lucru.

Generarea și utilizarea spațiilor de configurare permite reducerea problemei de optimizare cu șase dimensiuni la una bidimensională:

$$\max A(\vec{q}), \vec{q} \in R^2 \quad 4.8$$

cu suprafața dreptunghiulară

$$D(\vec{q}) = (q_{4max} - q_{4min}) * (q_{5max} - q_{5min}) \quad 4.9$$

și  $D(\vec{q}) \in S$ ,  $S \cong$  mulțimea de combinații de unghiuri de articulație  $q_4$  și  $q_5$  fără coliziuni.

Datorită proiectării robotului, unghiul de articulație  $q_1$  nu are nicio influență asupra problemei de optimizare deoarece se presupune că nu există niciun obstacol suplimentar în interiorul spațiului de lucru rotațional simetric al robotului.

Toate combinațiile posibile de unghiuri de articulație  $q_2$ ,  $q_3$ ,  $q_4$  și  $q_5$  sunt verificate pentru coliziuni cu o rezoluție de 10 grade. O dată ce sunt definite intervalele dorite de unghiuri de articulație  $q_4$  și  $q_5$ , spațiul de configurație al lui  $q_2$  și  $q_3$  poate fi generat rapid.

Optimizarea are ca rezultat următoarele intervale de unghiuri ale articulațiilor:

Tabel 4.2 Limitare intervale unghiuri articulații robot IRB 7600-500 pentru configurația optimizată

Robot IRB 7600- 500/2.55	Artic ulație	Limită min articulație $\vec{q}_{min}$	Limită max articulație $\vec{q}_{max}$	Limită min articulație $\vec{q}_{min,o}$	Limită max articulație $\vec{q}_{max,o}$
--------------------------------	-----------------	--	--	--	--

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

A	$q_1$	- 180 [°]	+ 180 [°]	- 180 [°]	+ 180 [°]
B	$q_2$	-60 [°]	+85 [°]	-60 [°]	+85 [°]
C	$q_3$	-180 [°]	+60 [°]	-180 [°]	+40 [°]
D	$q_4$	- 300 [°]	+ 300 [°]	- 175 [°]	+ 175 [°]
E	$q_5$	- 100 [°]	+100 [°]	- 90 [°]	+90 [°]
F	$q_6$	- 360 [°]	+ 360 [°]	- 360 [°]	+ 360 [°]

Optimizarea marginilor unghiurilor articulare este complicată, deoarece limitarea unui unghi articular are un efect important asupra limitelor altor unghiuri articulare.

Pe baza constrângerilor specifice aplicației pe care simulatorul trebuie să le îndeplinească, au fost derivate condițiile secundare ale unghiurilor articulare  $q_3$ ,  $q_4$  și  $q_5$ . Unghiul articular  $q_1$  nu a avut niciun efect asupra acestei investigații. Aceste condiții secundare au permis reducerea problemei de optimizare cu șase dimensiuni la o problemă de optimizare bidimensională. Pe baza spațiilor de configurație  $q_4$  și  $q_5$ , spațiul de lucru cartezian fără coliziune a fost investigat printr-o variație de intervale de unghiuri articulare ale lui  $q_4$  și  $q_5$ .

### 5. Modelarea matematică a algoritmului de indicare a mișcării

Scopul algoritmilor de indicare a mișcării este de a produce indici de mișcare care conduc spre percepția umană. Un algoritm de indicare a mișcării (MCA) utilizează coordonarea înclinării pentru a reproduce efectul mișcării de accelerație translațională susținută prin intermediul vectorului gravitațional, astfel încât să copieze percepția umană detectată de sistemul vestibular.

#### 5.1. Noțiuni introductive

Filtrul de constrângere este un algoritm de mișcare care transformă accelerația și viteza unghiulară a unui vehicul simulat în mișcare a platformei de simulator. Mișcarea produsă trebuie să se desfășoare în limitele, constrângerile și restricțiile platformei de mișcare utilizată pentru simulatorul de zbor, după cum a fost prezentat în capitolul anterior.

Filtrul de constrângere optimal utilizat pornește de la patru ipoteze:

- sistemul vestibular domină percepția indicilor de mișcare într-un simulator de zbor;
- discrepanța dintre indicii de mișcare dintre aeronava reală și din simulatorul cu mișcare poate fi măsurată prin valoarea medie pătratică a erorii vestibulare;
- mișcarea reală a aeronavei poate fi modelată ca un proces aleatoriu, nu depinde de constrângerile platformei de mișcare;
- sistemele dinamice, inclusiv sistemele vestibulare, pot fi reprezentate prin ecuații liniarizate.

Structura algoritmului prezentată în Figura 2.1 include două canale separate care generează percepția mișcării între o aeronavă reală și un simulator de zbor. În ambele canale, sistemul vestibular reprezintă sistemul vestibular al pilotului. În ceea ce privește cazul ideal, ieșirea simulatorului este identică cu intrarea.

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Scopul acestui algoritm este să determine o funcție de transfer  $W(s)$  constând în filtre trece-sus și filtre trece-jos.  $W(s)$  filtrează intrarea  $u_{A/C}$  a avionului pentru a obține datele de intrare ale simulatorului  $u_{sim}$  astfel încât să se minimizeze funcția de cost care conține eroarea de senzație a pilotului.

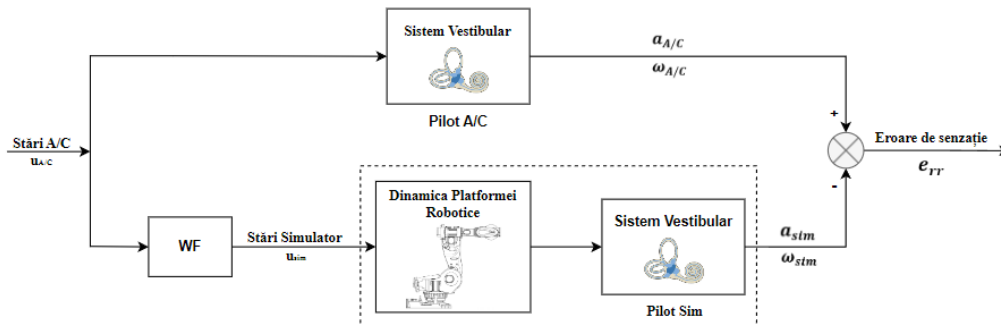


Figura 5.1 Structură generală algoritmului de control optimal, adaptare după [39]

În Figura 5.1 este prezentată tehnica de control optimal. Funcția obiectiv a acestui algoritmului include eroarea de percepție și constrângerile legate de mișcarea în spațiu de lucru al platformei de mișcare a simulatorului de zbor, unde  $a_{A/C}$  și  $a_{sim}$  reprezintă accelerația detectată în poziția reală și în poziția spațiului de lucru. De asemenea,  $\omega_{A/C}$  și  $\omega_{sim}$  sunt vitezele de rotație detectate în aceste poziții. Datorită utilizării unui model cu două intrări și două ieșiri, sunt utilizate patru funcții de transfer ( $W_{11}$ ,  $W_{21}$ ,  $W_{12}$  și  $W_{22}$ ), fiecare dintre acestea reprezentând efectul fiecărei intrări asupra fiecărei ieșiri. Pentru a se lua în considerare structura și parametrii algoritmului de mișcare este necesară convertirea intrării efective ( $u_{A/C}$ ) și a intrării simulatorului de zbor ( $u_{sim}$ ). Conform Figura 5.1 sunt două moduri diferite de comparare a percepției mișcării reale cu cea din simulatorul de zbor.

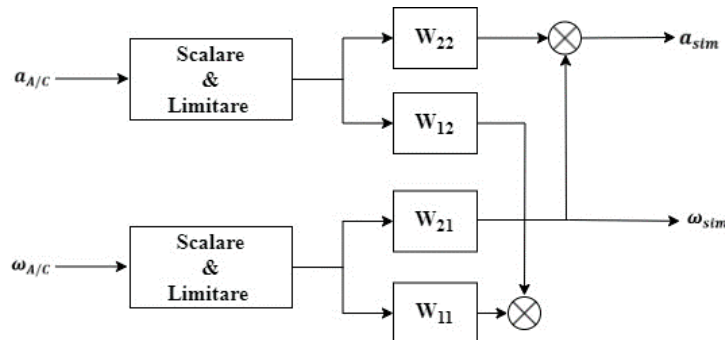


Figura 5.2 Algoritm filtru de constrângere optimal (OWF), adaptare după [17]

### 5.2. Scalare și limitare

Platforma de mișcare a simulatorului are limitări fizice, motiv pentru care dinamica sa trebuie să fie limitată în cadrul acestor limitări și constrângeri. Prin urmare, în cadrul schemei de funcționare a algoritmului de mișcare a fost aplicată pentru canalul translațional și rotațional o limitare, și totodată o scalare pentru semnalele de intrări ale modelului.

Limitarea și scalarea se aplică atât semnalelor de intrare translaționale  $a_{A/C}$ , cât și semnalelor de intrare rotaționale  $\omega_{A/C}$ . Limitarea și scalarea modifică amplitudinea semnalului de intrare în mod uniform pe toate frecvențele. Limitarea este un proces neliniar care reduce

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

semnalul astfel încât acesta să fie limitat la o valoare mai mică decât o anumită amplitudine. Limitarea și scalarea sunt utilizate pentru a reduce răspunsul la mișcare al unui simulator de zbor. Un polinom de ordinul trei a fost utilizat pentru scalare și a fost implementat în schema generală a algoritmului de mișcare. Atunci când magnitudinea intrării în sistemul de mișcare al simulatorului este mică, se dorește ca amplitudinea să fie relativ mare, altfel ieșirea să fie sub pragul de percepție al pilotului. Iar atunci când magnitudinea de intrare este mare, se dorește ca amplitudinea să fie relativ scăzută pentru a evita cazul în care simulatorul să încerce să depășească limitele hardware.

Pentru implementarea procedurii a fost notată intrarea cu  $x_a$ , respectiv ieșirea cu  $x_z$ . Apoi s-a definit  $x_{amax}$  ca intrare maximă dorită și  $x_{zmax}$  ieșire maximă, iar  $s_0$  și  $s_1$  pantele la  $x_a = 0$  și respectiv  $x_a = x_{amax}$ .

Sunt definite patru caracteristici dorite pentru scalarea neliniară astfel:

$$\begin{aligned} x_a = 0 &\Rightarrow x_z = 0 \\ x_a = x_{amax} &\Rightarrow x_z = x_{zmax} \\ x'_z|_{x_a=0} &= s_0 \\ x'_z|_{x_a=x_{amax}} &= s_1 \end{aligned} \quad 5.1$$

Polinomul de ordinul trei utilizat pentru a furniza funcții cu caracteristicile specifice este de forma:

$$x_z = p_3 x_a^3 + p_2 x_a^2 + p_1 x_a^1 + p_0 \quad 5.2$$

unde

$$\begin{aligned} p_0 &= 0 \\ p_1 &= s_0 \\ p_2 &= x_{amax}^{-2} (3 \cdot x_{zmax} - 2 \cdot s_0 \cdot x_{amax} - s_1 \cdot x_{amax}) \\ p_3 &= x_{amax}^{-3} (s_0 \cdot x_{amax} - 2 \cdot x_{zmax} + s_1 \cdot x_{amax}) \end{aligned} \quad 5.3$$

Scalarea parametrilor pentru canalul translațional:

$$\max|x_{amax}| = 6 \text{ m/s}^2 \quad 5.4$$

$$\max|x_{zmax}| = 6 \frac{\text{m}}{\text{s}^2}, \text{ pentru } X \quad 5.5$$

$$\max|x_{zmax}| = 0.8 \frac{\text{m}}{\text{s}^2}, \text{ pentru } Y \text{ și } Z \quad 5.6$$

iar coeficienții sunt:

$$\begin{aligned} s_0 &= 1, \text{ pentru } X \\ s_0 &= 0.2, \text{ pentru } Y \text{ și } Z \\ s_1 &= 0.1, \text{ pentru } X, Y \text{ și } Z \end{aligned} \quad 5.7$$

Scalarea parametrilor pentru canalul rotațional:

$$\max|x_{amax}| = 3.14 \frac{\text{rad}}{\text{s}} \quad 5.8$$

$$\max|x_{zmax}| = 1.57 \frac{\text{rad}}{\text{s}} \quad 5.9$$

iar coeficienții sunt:



$$\begin{aligned} s_0 &= 0.785, \text{ pentru } X, Y \text{ și } Z \\ s_1 &= 0.1, \text{ pentru } X, Y \text{ și } Z \end{aligned} \tag{5.10}$$

Accelerațiile liniare sunt limitate pe toate axele la  $6 \frac{m}{s^2}$  pentru canalul translațional, iar limitarea pentru canalul rotațional este de  $3.14 \frac{rad}{s}$ . Pentru a calcula coeficienții de scalare polinomială pentru fiecare grad de libertate al aeronavei pentru intrarea în algoritmul de mișcare optimal se implementează diagrama de flux prezentată în Figura 5.3

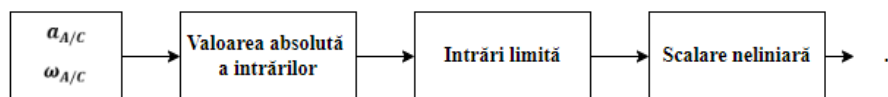


Figura 5.3 Diagrama de flux pentru limitare și scalare

### 5.3. Modelul matematic al sistemului vestibular

Conform Figura 5.1 modelul matematic de implementare a algoritmului de indicare a mișcării cuprinde un modul care încorporează modelul matematic pentru sistemul vestibular. Acest modul este esențial în implementarea algoritmului de filtrare optimal. Pentru o bună implementare a modelelor matematice pentru cele două elemente ale sistemului vestibular, canalul semicircular și organul otolitic, se realizează o cercetare privind pragurile de mișcare pentru a defini valorile specifice ce vor fi utilizate în dezvoltarea algoritmului de indicare a mișcării.

Sistemul vestibular (Figura 5.4) este situat în urechea internă și constă din canalele semicirculare care detectează mișcarea unghiulară, respectiv organele otolitice care detectează mișcarea liniară [53].

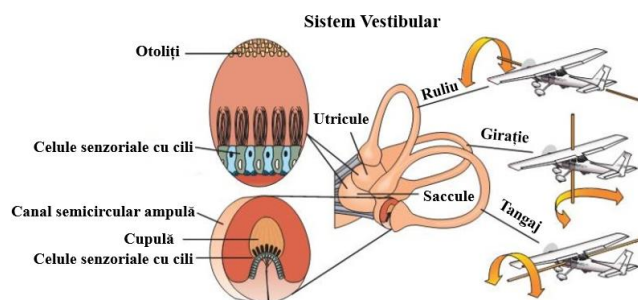


Figura 5.4 Relația dintre sistemul vestibular și percepția umană, adaptare după [54]

Un sistem vestibular include două grade de libertate, supraîncărcare și tangaj după cum este reprezentat în Figura 5.5, unde  $a_{Ax}$  reprezintă accelerația liniară, iar  $\dot{\theta}$  reprezintă viteza rotațională. Ca ieșiri din modelul vestibular avem  $a_{Ax_s}$  care reprezintă accelerația simțită, și  $\dot{\theta}_s$  care reprezintă viteza rotațională simțită. Vectorul  $g$  reprezintă accelerația gravitațională, iar semnul integralei definește integrala vitezei rotaționale față de poziția de deplasare.

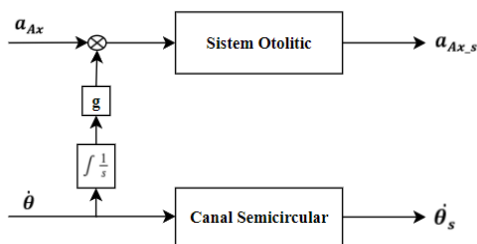


Figura 5.5 Model sistem vestibular pentru percepția umană

Modelarea matematică a sistemului vestibular pentru canalul semicircular și organul otolic a fost implementată în MATLAB / Simulink după cum este prezentat în Figura 5.6, fiind incorporate în modelul acestuia modelele prezentate în Figura 5.1, Figura 5.2 și Figura 5.5.

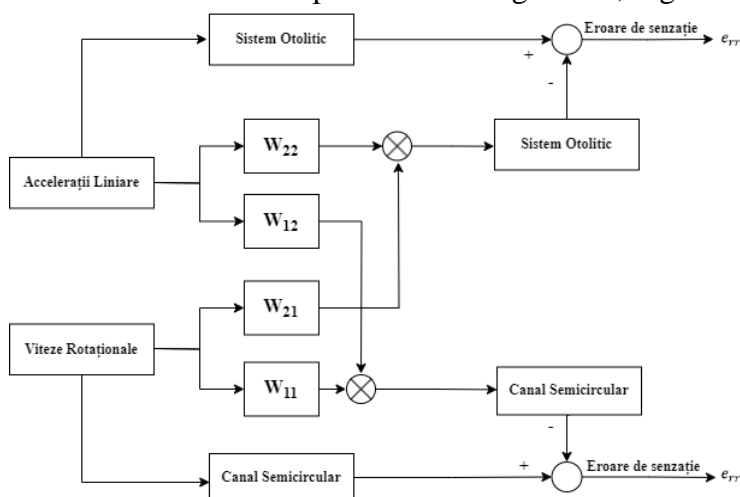


Figura 5.6 Model sistem vestibular cu filtru de constrângere integrat

Scopul algoritmului filtru de constrângere optimal bazat pe LQR este de a determina o funcție de transfer  $W(s)$  care relaționează intrarea în mișcare a simulatorului de zbor ( $u_{sim}$ ) cu intrarea în mișcare a aeronavei ( $u_{A/C}$ ).

$$u_{sim}(s) = W(s) \cdot u_{A/C}(s) \tag{5.11}$$

Intrările de control, care cuprind accelerațiile și unghiurile Euler, sunt aplicate pentru a produce comenzile de baza ale mișcării dorite. Astfel algoritmul filtru de constrângere optimal bazat pe LQR determină accelerația simulatorului prin minimizarea erorii de senzație umană între simulator și aeronavă, precum și a mișcării liniare și unghiulare a platformei. Astfel scopul algoritmului este de a limita eroarea de senzație umană și mișcarea platformei în limitele spațiului fizic de lucru al platformei robotice. Inputul  $u$  este reprezentat astfel:

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ a_{Ax} \end{bmatrix} \tag{5.12}$$

Un pas important în implementarea algoritmului este modul de definire a efectului de coordonare a înclinării. Coordonarea înclinării are ca scop calcularea componentelor de frecvență joasă pe axa de translație orizontală, direcția X, și pe axa de translație laterală, direcția Y, iar componentele de frecvență joasă pe axa de translație verticală, direcția Z nu se calculează. Figura 5.7 prezintă cum se formulează coordonarea înclinării pentru a genera accelerație pe axele orizontală și laterală.

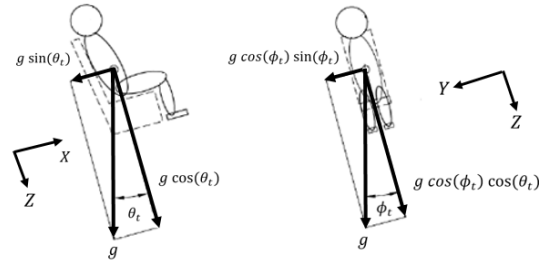


Figura 5.7 Formulare coordonare a înclinării

Forța specifică de supraîncărcare în mișcarea centrală a simulatorului este obținută prin ecuația de mai jos astfel:

$$f_s = a_{Ax} \cdot \cos(\theta_t) + g \cdot \sin(\theta_t) \cong a_{Ax} + g \cdot \theta_t \quad 5.13$$

În aproximarea finală a ecuației 5.13 s-au luat în considerare aproximarea pentru unghiuri mici, astfel  $\sin(\theta_t)$  și  $\cos(\theta_t)$  fiind înlocuite cu  $\theta_t$  și respectiv 1.

Organul otolitic detectează mișcarea liniară în urechea internă. Meiry și colaboratorii săi [55] au furnizat un model modificat pentru indicarea accelerației liniare, precum și pentru indicarea înclinării pe baza cercetării din lucrarea [37].

Valoarea specifică detectată  $a_{Ax_s}$  este legată de forța specifică a stimulului detectată de modelul otolitic și este introdusă în ecuația de mai jos astfel:

$$a_{Ax_s} = k_o \frac{s + A_0}{(s + B_0)(s + B_1)} \cdot f_s \quad 5.14$$

Aplicând transformata Laplace, o nouă formă a ecuației poate fi obținută. Termenul  $\frac{1}{s}$  este integrarea vitezei unghiulare.

$$f_s(s) = a_{Ax}(s) + g \cdot \frac{1}{s} \cdot \dot{\theta}(s) \quad 5.15$$

Substituind ecuația 5.13 în ecuația 5.14 va rezulta următoarea ecuație astfel:

$$a_{Ax_s} = k_o \frac{s + A_0}{(s + B_0)(s + B_1)} \left( a_x(s) + g \cdot \frac{1}{s} \cdot \dot{\theta}(s) \right) \quad 5.16$$

unde  $A_0$ ,  $B_0$  și  $B_1$  reprezintă parametrii forței specifice din modelul otolitic.

Ecuația 5.16 poate fi rescrisă astfel:

$$a_{Ax_s} + a \cdot a_{Ax_s} + b \cdot a_{Ax_s} = c \cdot u_1 + d \cdot u_1 + e \cdot \int u_1 dt + f \cdot u_2 + h \cdot u_2 \quad 5.17$$

Reprezentarea în spațiul stărilor este dată de următoarea formă astfel:

$$\begin{aligned} \dot{x}_{ot} &= A_{ot} \cdot x_{ot} + B_{ot} \cdot u \\ a_{Ax_s} &= C_{ot} \cdot x_{ot} + D_{ot} \cdot u \end{aligned} \quad 5.18$$

unde  $x_{ot}$  reprezintă stările pentru modelul otolitic și  $u$  reprezintă intrarea sistemului vestibular, iar  $A_{ot}$ ,  $B_{ot}$ ,  $C_{ot}$  și  $D_{ot}$  sunt definite astfel:

$$A_{ot} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -b & -a & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -b & -a \end{bmatrix} \quad 5.19$$

$$B_{ot} = \begin{bmatrix} c & 0 \\ d - a \cdot c & 0 \\ e & 0 \\ 0 & f \\ 0 & h - a \cdot f \end{bmatrix} \quad 5.20$$

$$C_{ot} = [1 \ 0 \ 0 \ 1 \ 0] \quad 5.21$$

$$D_{ot} = [0 \ 0] \quad 5.22$$

Viteza unghiulară detectată  $\dot{\theta}_s$  este legată de viteza unghiulară reală detectată de sistemul semicircular. Canalele semicirculare pot detecta senzația de mișcare de rotație. Telban și Cardullo [56] au propus o funcție de transfer a modelului canalelor semicirculare pentru a detecta senzația de viteză unghiulară atunci când aceasta este aplicată ca stimul. Această funcție de transfer conține modelul pendulului de torsiune, operatorul de adaptare și un termen de plumb. Această funcție oferă cea mai bună aproximare a dinamicii reale a sistemului vestibular de senzație pentru mișcarea de rotație. Modelul pentru canalul semicircular este definit astfel:

$$\dot{\theta}_s = \frac{k_{sc} \cdot \tau_1 \cdot \tau_a \cdot s^2 (1 + \tau_L \cdot s)}{(\tau_1 \cdot s + 1) + (\tau_2 \cdot s + 1) + (\tau_a \cdot s + 1)} \dot{\theta} \quad 5.23$$

unde  $\tau_1$ ,  $\tau_2$ ,  $\tau_L$  și  $\tau_a$  sunt constanta de timp lung, constanta de timp scurt, termenul de plumb, și constanta operatorului de adaptare.

Ecuția 5.23 poate fi rescrisă astfel:

$$\dot{\theta}_s = \frac{T_5 \cdot s^3 + T_4 \cdot s^2}{s^3 + T_3 \cdot s^2 + T_2 \cdot s + T_1} \dot{\theta} \quad 5.24$$

unde  $T_0$ ,  $T_1$ ,  $T_2$ ,  $T_3$  și  $T_4$  sunt constante și pot fi definite astfel:

$$T_1 = \frac{1}{(\tau_1 \cdot \tau_2 \cdot \tau_a)} \quad 5.25$$

$$T_2 = \frac{(\tau_1 + \tau_2 + \tau_a)}{(\tau_1 \cdot \tau_2 \cdot \tau_a)} \quad 5.26$$

$$T_3 = \frac{(\tau_1 \cdot \tau_a + \tau_1 \cdot \tau_2 + \tau_a \cdot \tau_2)}{(\tau_1 \cdot \tau_2 \cdot \tau_a)} \quad 5.27$$

$$T_4 = \frac{k_{sc}}{\tau_2} \quad 5.28$$

$$T_5 = \frac{k_{sc} \cdot \tau_L}{\tau_2} \quad 5.29$$

Funcția de transfer din ecuația 5.24 poate fi exprimată în spațiul stărilor astfel:

$$\begin{aligned} \dot{x}_{sc} &= A_{sc} \cdot x_{sc} + B_{sc} \cdot u \\ \dot{\theta}_s &= C_{sc} \cdot x_{sc} + D_{sc} \cdot u \end{aligned} \quad 5.30$$

unde matricele  $A_{sc}$ ,  $B_{sc}$ ,  $C_{sc}$  și  $D_{sc}$  sunt definite astfel:

$$A_{sc} = \begin{bmatrix} -T_3 & 1 & 0 \\ -T_2 & 0 & 1 \\ -T_1 & 0 & 0 \end{bmatrix} \quad 5.31$$

$$B_{sc} = \begin{bmatrix} T_4 - T_3 \cdot T_5 & 0 \\ -T_2 \cdot T_5 & 0 \\ -T_1 \cdot T_5 & 0 \end{bmatrix} \quad 5.32$$

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

$$C_{sc} = [1 \ 0 \ 0] \quad 5.33$$

$$D_{sc} = [T_5 \ 0] \quad 5.34$$

Reprezentarea spațiului de stare a sistemului vestibular uman, care constă atât din canalul semicircular, cât și din organul otolitic este definită astfel:

$$\begin{aligned} \dot{x}_{SV} &= A_{SV} \cdot x_{SV} + B_{SV} \cdot u \\ y_S &= C_{SV} \cdot x_{SV} + D_{SV} \cdot u \end{aligned} \quad 5.35$$

unde  $x_{SV}$  conține stările modelului vestibular uman, iar  $y_S$  este răspunsul detectat. Matricele  $A_{SV}$ ,  $B_{SV}$ ,  $C_{SV}$  și  $D_{SV}$  sunt reprezentate astfel:

$$A_{SV} = \begin{bmatrix} A_{sc} & 0 \\ 0 & A_{ot} \end{bmatrix} \quad 5.36$$

$$B_{SV} = \begin{bmatrix} B_{sc} \\ B_{ot} \end{bmatrix} \quad 5.37$$

$$C_{SV} = \begin{bmatrix} C_{sc} & 0 \\ 0 & C_{ot} \end{bmatrix} \quad 5.38$$

$$D_{SV} = \begin{bmatrix} D_{sc} \\ D_{ot} \end{bmatrix} \quad 5.39$$

Eroarea de stare a sistemului vestibular și eroarea de senzație a pilotului sunt definite astfel:

$$\begin{aligned} \dot{x}_{err} &= A_{SV} \cdot x_{err} + B_{SV} \cdot u_{sim} - B_{SV} \cdot u_{A/C} \\ err &= C_{SV} \cdot x_{err} + D_{SV} \cdot u_{sim} - D_{SV} \cdot u_{A/C} \end{aligned} \quad 5.40$$

unde intrările simulatorului și ale aeronavei sunt  $u_{sim}$  respectiv  $u_{A/C}$ .

### 5.4. Procedura de implementare a algoritmului de indicare a mișcării

Vectorul semnalului de intrare este considerat ca fiind:

$$u_{A/C}(s) = [a_{Ax}(s) \ \dot{\theta}(s)] \quad 5.41$$

Prin completarea ecuațiilor și derivarea noilor termeni, se generează o funcție de transfer generalizată, care leagă mișcarea reală cu mișcarea simulatorului de zbor.

$$u_{sim}(s) = W(s) \cdot u_{A/C}(s) \quad 5.42$$

unde  $W$  este matricea funcției de transfer optimizate care transferă intrările simulatorului  $u_{sim}(s)$  la mișcarea reală  $u_{A/C}(s)$ . Funcția de transfer  $W(s)$  este definită sub formă de matrice astfel:

$$W(s) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \quad 5.43$$

Forma fiecărei funcții de transfer  $W_{ij}$  este prezentată în ecuația 5.44, în total, fiind utilizați doisprezece parametri pentru completarea numitorului și, respectiv, a numărătorului. Parametrii numitorului sunt unici pentru toate  $W_{ij}$ .

$$W_{ij} = \frac{a_5 \cdot s^5 + a_4 \cdot s^4 + a_3 \cdot s^3 + a_2 \cdot s^2 + a_1 \cdot s + a_0}{b_5 \cdot s^5 + b_4 \cdot s^4 + b_3 \cdot s^3 + b_2 \cdot s^2 + b_1 \cdot s + b_0} \quad 5.44$$

Găsirea matricei corespunzătoare pentru soluția numerică nu este simplă și poate fi un eșec, din acest motiv, pentru identificarea parametrilor optimi s-a utilizat algoritmul genetic.

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Funcția obiectiv pentru această problemă este prezentată în ecuația de mai jos astfel:

$$F_c = \int_0^{t_f} [|z_{sim} - z_0| + |p_{A/C} - p_{sim}|] dt \quad 5.45$$

Poziția de ridicare a simulatorului de zbor este notată cu  $z_{sim}$  și valoarea de bază, constrângerea fizică a simulatorului de zbor este notată cu  $z_0$ . Iar  $p_{A/C}$  este o funcție de percepție sesizată (percepția umană) înainte și după aplicarea filtrului de constrângere. Notarea  $A/C$  și  $sim$  denotă mișcarea reală și, respectiv, mișcarea simulatorului. Timpul final din ecuație este definit de  $t_f$ .

În Figura 5.8 este prezentată o imagine de ansamblu a modelului implementat. Diagrama prezintă procedura și etapele generale ale modului de implementare a filtrului de constrângere optimal. Utilizând elementele de bază ale filtrului de constrângere clasic și funcțiile de transfer (ecuația 5.44), se pregătește baza acestui filtru, apoi se ia în considerare funcția de cost (ecuația 5.45) pentru a reduce eroarea de percepție și a exclude simulatorul de zbor din constrângerea sa fizică, iar apoi cu ajutorul algoritmului de optimizare, se vor obține coeficienții funcțiilor de filtrare.

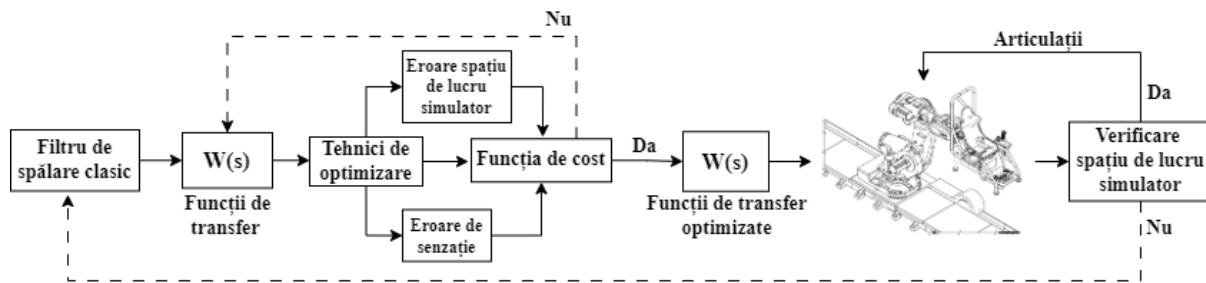


Figura 5.8 Procedura de implementare a algoritmului de indicare a mișcării bazat pe OWF

După cum am menționat mai sus scopul algoritmului de indicare al mișcării optimal bazat pe regulator liniar pătratic (LQR) este determinarea unei funcții de transfer,  $W(s)$ , care să pună în relație intrarea dată de mișcare simulatorului,  $u_{sim}$ , cu intrarea dată de mișcarea avionului,  $u_{A/C}$  conform ecuației 5.42. Pentru a produce comenzile de bază ale mișcării dorite sunt aplicate intrările de control și anume accelerațiile liniare și vitezele unghiulare. Metoda algoritmului de mișcare optimal bazată pe LQR determină accelerația simulatorului prin minimizarea erorii de senzație umană între simulator și aeronava reală, precum și a mișcării liniare și unghiulare a platformei. Scopul metodei este de a limita eroarea de senzație umană și mișcarea platformei în conformitate cu limitările spațiului de lucru al platformei de mișcare – sistem robotic serial.

### 5.5. Algoritm genetic

Algoritmii evolutivi (AE) sunt metode robuste de căutare a soluțiilor aproape optime, cu capacitatea de a gestiona evaluări nedefinite care au anumite proprietăți, cum ar fi: cuplarea; variația în timp; discontinuitatea; zgomotul; și probabilitatea [57]. Algoritmul genetic (AG) este unul dintre algoritmii evolutivi eficienți, fiind o metodă de căutare euristică, robustă și fiabilă care generează o soluție printr-un proces care imită selecția și evoluția naturală. AG este un algoritm stocastic bazat pe populație, iar potențialul algoritmului de a rezolva probleme complicate este un avantaj important al acestuia comparativ cu tehnicile clasice de optimizare.

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Fiecare cromozom din populația AG caută în spațiul de soluții în mod aleatoriu, independent unul de celălalt și, în același timp, în direcții diferite. Acest lucru face ca algoritmul să fie ideal pentru implementarea paralelă [58]. AG utilizează operații genetice (selecție, încrucișare și mutație) pentru a face să evolueze soluțiile. În AG, o populație cuprinde un set de cromozomi individuali [59]. Procesul de evoluție prin crearea unei noi populații se repetă până când este îndeplinit un criteriu de oprire predefinit. Astfel algoritmul este aplicat pentru a îmbunătăți soluția oferită de algoritmul filtru de constrângere optimal bazat pe LQR. Scopul este de a îmbunătăți percepția umană, îmbunătățirea urmăririi formei semnalului și, în același timp, reducerea indicilor falși de mișcare. Metoda ia în considerare mai mulți factori, precum limitările simulatorului, criteriul de urmărire a formei, acuratețea semnalului și pragul uman în coordonarea înclinării. În Figura 5.9 este ilustrată schema operatorilor AG pentru optimizarea algoritmului de mișcare ce urmează să fie explicată în continuare.

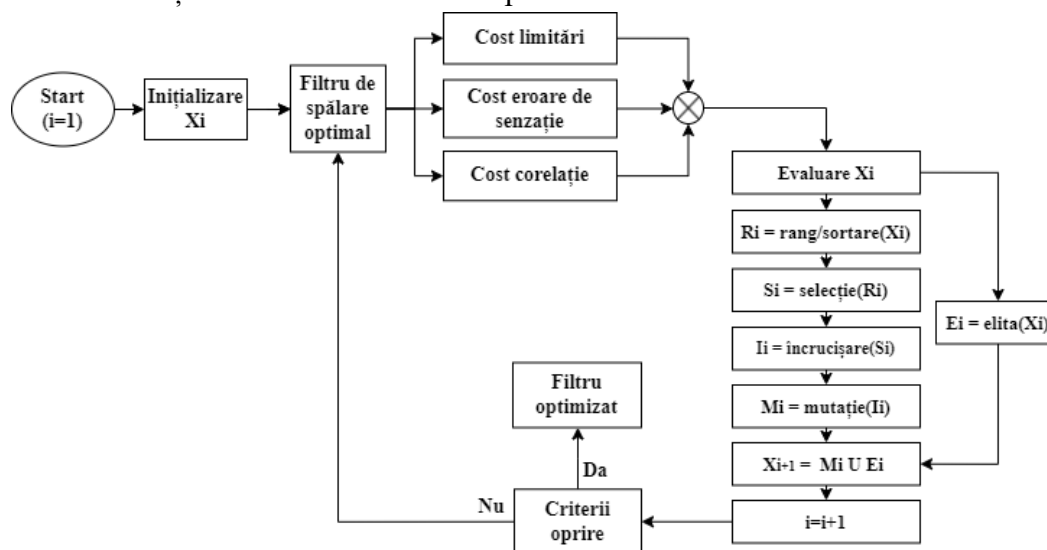


Figura 5.9 Structura algoritmului genetic

Fiecare soluție la a-i-a generație este reprezentată de un cromozom format din de gene:

$$A_c^i = [A_{c,1}^i, A_{c,2}^i, \dots, A_{c,n}^i] \quad 5.46$$

unde  $c$  este numărul de cromozomi din populație și  $n$  este numărul de gene din fiecare cromozom. Pentru a evalua fiecare individ, este necesar să se stabilească un indice de performanță definit. Funcția obiectiv este formulată pentru a clasifica performanța fiecărui individ. Pentru metoda propusă, funcția obiectiv globală cuprinde cinci subprobleme, astfel:

- I. Minimizarea erorilor de translație și de rotație ale senzației umane simțită de piloți în aeronavă reală și în simulator.
- II. Minimizarea deplasărilor unghiulare și liniare ale platformei de mișcare pentru a funcționa în limitele sale fizice.
- III. Minimizarea fluctuației erorilor de percepție.
- IV. Maximizarea coeficientului de corelație, definit ca fiind coeficientul de corelație de moment Pearson, pentru a crește gradul de urmărire a formei semnalului de referință în semnalul produs.
- V. Minimizarea accelerației și a vitezei platformei de mișcare datorită constrângerilor fizice.

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Prin urmare, funcția obiectiv la fiecare a-i-a iterație pentru al-c-lea individ este definită ca suma funcțiilor subiectiv:

$$O(A_c^i) = O_{c\_sener}^i + O_{c\_cor}^i + O_{c\_lim}^i \quad 5.47$$

Cromozomii din AG sunt inițializați utilizând indivizii obținuți din rezultatele algoritmului filtru de constrângere optimal bazat pe LQR.

Erorile de senzație umană liniară și rotațională între pilotul real și pilotul din simulator se definesc astfel:

$$er_a(t) = a_{A/c}(t) - a_{sim}(t) \quad 5.48$$

$$er_{\dot{\theta}}(t) = \dot{\theta}_{A/c}(t) - \dot{\theta}_{sim}(t) \quad 5.49$$

Pentru a defini eroarea de senzație umană între senzațiile reale și simulate, în funcția obiectiv se utilizează un indice de performanță a integralei erorii pătrate. Funcțiile obiectiv aferente sunt definite astfel:

$$O_{c\_liner}^i = k_{a\_er} \int_{t_i}^{t_f} er_a^2(t) dt \quad 5.50$$

$$O_{c\_roter}^i = k_{\dot{\theta}\_er} \int_{t_i}^{t_f} er_{\dot{\theta}}^2(t) dt \quad 5.51$$

unde  $k_{a\_er}$  și  $k_{\dot{\theta}\_er}$  sunt ponderile care determină influența erorilor de senzație de translație și de rotație în funcția de cost, iar  $t_i$  reprezintă timpul de începere și  $t_f$  timpul de terminare test. Aceste ponderi sunt alese în conformitate cu cercetările anterioare ale modelelor algoritmilor de mișcare [56]. Funcția obiectiv care se referă la eroarea totală a senzației umane este definită ca fiind suma funcțiilor obiectiv de translație și rotație:

$$O_{c\_sener}^i = O_{c\_liner}^i + O_{c\_roter}^i \quad 5.52$$

În continuare este definită funcția obiectiv a coeficientului de corelație încrucișată care indică dependența a două semnale. Un coeficient de corelație crescut permite semnalului generat să urmeze semnalul de senzație reală cu precizie. Coeficientul de corelație Pearson între două semnale  $y_1(t)$  și  $y_2(t)$  este definit astfel:

$$C_c(y_1, y_2) = \frac{C_v(y_1, y_2)}{\sqrt{C_v(y_1, y_1) \cdot C_v(y_2, y_2)}} \quad 5.53$$

Dar poate fi exprimat sub o altă formă astfel:

$$C_c(y_1, y_2) = \frac{C_v(y_1, y_2)}{\sigma_{y_1} \cdot \sigma_{y_2}} \quad 5.54$$

Covarianța a două semnale este:

$$C_v(y_1, y_2) = E[(y_1 - E[y_1])(y_2 - E[y_2])] \quad 5.55$$

$$\sigma_{y_i} = \sqrt{E[(y_i - E[y_i])^2]} \quad 5.56$$

unde  $E[y_i]$  reprezintă media semnalelor  $y_i$ ,  $\sigma_{y_i}$  reprezintă abaterea (deviația) standard a semnalului și  $E[...]$  reprezintă valoarea dorită.

Coeficientul de corelație dintre  $y_1(t)$  și  $y_2(t)$  este:



Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

$$C_c(y_1, y_2) = \frac{E[(y_1 - E[y_1])(y_2 - E[y_2])]}{\sigma_{y_1} \cdot \sigma_{y_2}} \quad 5.57$$

Astfel poate fi definită funcția obiectiv de corelație încrucișată pentru a oferi o soluție cu o aproximare mai bună a formei semnalului de senzație de referință astfel:

$$O_{c\_lincor}^i = k_{a\_cor} |1 - C_c(a_{A/C}, a_{sim})| \quad 5.58$$

$$O_{c\_rotcor}^i = k_{\theta\_cor} |1 - C_c(\theta_{A/C}, \theta_{sim})| \quad 5.59$$

unde  $k_{a\_cor}$  și  $k_{\theta\_cor}$  reprezintă ponderile liniare și unghiulare ale coeficientului de corelație.

Funcția obiectiv totală a coeficientului de corelație este definită astfel:

$$O_{c\_cor}^i = O_{c\_lincor}^i + O_{c\_rotcor}^i \quad 5.60$$

În ceea ce privește limitele platformei de mișcare, deplasările liniare și unghiulare trebuie să fie reduse la minim, pentru a evita atingerea limitei spațiului de lucru. Pentru a preveni atingerea unor viteze/accelerații liniare și a unor viteze unghiulare extreme în cadrul platformei de mișcare, funcția obiectiv de limitare trebuie să includă factori de viteză și accelerație.

Funcția obiectiv de limitare este reprezentată astfel:

$$O_{c\_linlim}^i = k_x \int_{t_i}^{t_f} x^2(t) dt + k_v \int_{t_i}^{t_f} v_{sim}^2(t) dt + k_a \int_{t_i}^{t_f} a_{sim}^2(t) dt \quad 5.61$$

$$O_{c\_rotlim}^i = k_{\theta} \int_{t_i}^{t_f} \theta^2(t) dt + k_{\dot{\theta}} \int_{t_i}^{t_f} \dot{\theta}_{sim}^2(t) dt \quad 5.62$$

unde  $k_x$  este o pondere pentru limitarea deplasării liniare, iar  $k_v$  și  $k_a$  sunt pentru limitările vitezei și accelerației liniare. În ecuația 5.62,  $k_{\theta}$  reprezintă o pondere pentru deplasarea unghiulară.

$$O_{c\_lim}^i = O_{c\_linlim}^i + O_{c\_rotlim}^i \quad 5.63$$

Aceste limitări au fost încorporate în blocul de simulare sub formă de constrângeri, pentru a restricționa mișcările în limitele specifice ale simulatorului.

Procesul de selecție AG este utilizat pentru a alege indivizii pentru reproducere, acordând o probabilitate mai mare indivizilor mai potriviți. Pentru fiecare evaluare a funcției obiectiv un model MATLAB/Simulink este simulat și rulat într-o buclă închisă peste generația AG, care are ca rol minimizarea funcției obiectiv. Valorile brute sunt scalate de operația de selecție, folosind scalarea rangului astfel:

$$Sc(f_i) = \frac{1}{\sqrt{R_i}} \quad 5.64$$

unde  $R$  este rangul indivizilor din populație, începând de la 1 pentru cel mai bun individ. Pentru a calcula funcția obiectiv, se aplică regula lui Simpson. În procesul de reproducere AG, se adoptă schema de selecție proporțională cu condiția fizică. Astfel, un cromozom cu o valoare obiectiv mai mare are o probabilitate mai mare de a fi copiat în generația următoare. În Figura 5.10 este prezentată schema de simulare pentru evaluarea performanței algoritmului filtru de constrângere optimal ajustat pe baza AG prin luarea în considerare a mai multor factori în funcția obiectiv după cum a fost menționat mai sus în prezentul subcapitol. Astfel operațiile AG, care includ selecția, încrucișarea și mutația din Figura 5.9, și funcțiile de transfer ale

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

filtrului de constrângere optimal sunt ajustate și simulate pentru a minimiza funcția obiectiv totală.

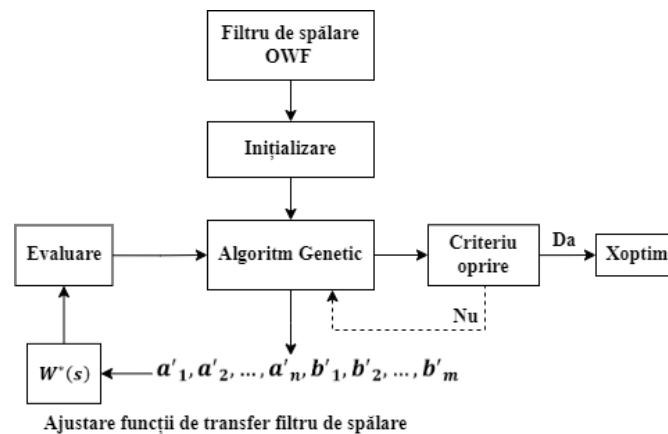


Figura 5.10 Structura algoritmului genetic pentru ajustarea funcțiilor de transfer ale filtrului de constrângere

Cu ajutorul optimizării AG, proiectarea unei funcții de transfer în buclă închisă prin selectarea coeficienților numărătorului și numitorului este un proces simplificat. Ca atare, filtrul de constrângere optimal este optimizat prin încorporarea unor aspecte noi și luarea în considerare a efectelor acestora. AG este utilizat pentru a ajusta informațiile de magnitudine și fază din funcțiile de transfer ale filtrului de constrângere optimal și pentru a îmbunătăți zerourile și polii din numitor și numărător. Acest lucru este util deoarece fiecare criteriu poate afecta fiecare parametru al funcțiilor de transfer ale filtrului de constrângere. Pentru a modifica polii și zerourile unei funcții de transfer, se manipulează coeficienții numărătorului și numitorului, după cum urmează:

$$W(s) = \frac{B(s)}{A(s)} = \frac{a_0 + a_1s^{-1} + \dots + a_ms^{-m}}{b_0 + b_1s^{-1} + \dots + b_ns^{-n}} \quad 5.65$$

unde  $s = j\omega$  reprezintă domeniul de frecvență,  $m$  este ordinul numărătorului,  $n$  este ordinul numitorului, iar  $a_i$  și  $b_i$  sunt coeficienții numărătorului și numitorului. Ajustarea acestora poate fi reprezentată astfel:

$$W^*(s) = \frac{B^*(s)}{A^*(s)} = \frac{a_0a'_0 + a_1a'_1s^{-1} + \dots + a_ma'_ms^{-m}}{b_0b'_0 + b_1b'_1s^{-1} + \dots + b_nb'_ns^{-n}} \quad 5.66$$

unde  $a'_i$  și  $b'_i$  sunt coeficienți pentru manipularea coeficienților primari ai numărătorului și numitorului, iar  $a'_0 = b'_0 = 1$  pentru eliminarea erorii în regim staționar. Apoi funcția de transfer parametrizată este evaluată prin intermediul funcției obiectiv care urmează să fie optimizată prin ajustarea coeficienților.

Fiecare soluție poate fi reprezentată ca un vector

$$X = [a'_1, a'_2, \dots, a'_n, b'_1, b'_2, \dots, b'_m] \quad 5.67$$

unde  $X$  reprezintă cromozomul AG, iar  $a_i$  și  $b_i$  sunt gene înrudite.

Scopul optimizării AG este de a găsi o soluție pentru minimizarea funcției obiectiv din ecuația 5.47. Deplasarea zerourilor și a polilor ia în considerare coeficientul de corelație, limitările erorii de percepție, precum și neliniaritățile care nu au fost implementate în metoda LQR. Aceste aspecte noi sunt încorporate în funcția obiectiv pentru a găsi o soluție mai bună

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

pentru filtrele de constrângere optimale existente. O diagramă schematică a evaluării ajustării AG pentru dezvoltarea filtrului de constrângere bazat pe AG este prezentată în Figura 5.10.

Prin urmare, soluția bazată pe filtrul de constrângere optimal constituie indivizii inițiali care trebuie luați în considerare de către AG.

$$(X_0^{int})_{1 \times (m+n)} = X_n = [1, 1, \dots, 1] \quad 5.68$$

Prin schimbarea numărătorului și numitorului, zerourile și poli se deplasează de la valorile lor inițiale.

Rata de schimbare pentru toți factorii este luată în considerare după cum urmează:

$$\begin{aligned} r \leq a'_i &\leq \frac{1}{r} \\ r \leq b'_i &\leq \frac{1}{r} \\ rX_n &\leq X_i^t \leq X_n \end{aligned} \quad 5.69$$

unde  $r$  este o rată arbitrară mai mică decât 1, dar care nu trebuie să fie foarte departe de această valoare. Selectarea lui  $r$  depinde de cazul studiat, iar în prezentul studiu, după mai multe încercări, rata a fost stabilită la 0,81. Această setare garantează că noii poli și zerouri rămân foarte aproape de soluțiile anterioare din algoritmul filtru de constrângere optim, chiar și în cazul celei mai proaste soluții.

Ecuția 5.69 determină intervalele de parametri utilizate în procesul de căutare AG pentru a rezolva problema de optimizare. Fiecare set de parametri obținut pentru filtrul de constrângere este evaluat prin procesul de simulare. Bucla de evoluție AG continuă să caute până când este îndeplinit criteriul de oprire, care reprezintă un prag pentru valoarea funcției obiectiv.

## 6. Rezultate

În prezentul capitol sunt evidențiate rezultatele specifice pentru algoritmul filtru de constrângere optimal de indicare a mișcării formulat, proiectat, dezvoltat, implementat, testat și validat pentru îmbunătățirea mediului de simulare a zborului prin utilizarea simulatorului de zbor bazat pe sistem robotic serial de tip IRB 7600-500. În vederea validării aplicabilității filtrului de constrângere optimal și compararea cu filtrul modificat pe baza informațiilor oferite de algoritmul genetic aplicat pentru ajustarea filtrului de constrângere optimal inițial au fost realizate șase zboruri diferite, fiind utilizat un model dinamic pentru o aeronavă de evoluție rapidă. Așa cum e prezentat în capitolul anterior filtru de constrângere optimal bazat pe LQR este implementat având centrul de rotație al platformei simulatorului în zona capului subiectului (pilotului) cu scopul de a elimina potențiali indici de mișcare falși. Ca intrări ale filtrului de constrângere sunt considerate accelerația liniară și viteza unghiulară. Totodată modelul sistemului vestibular cu canalul semicircular și organul otolitic sunt parte din schema filtrului pentru a produce un filtru de constrângere adecvat nevoii de eliminare a indicilor falși de mișcare. Parametrii pentru sistemul vestibular definiți în capitolul 5 pentru sistemul otolitic și canalul semicircular sunt prezentați în tabelul de mai jos, conform [56].

În Tabel 6.1 sunt prezentați parametrii forței specifice pentru sistemul otolitic, iar răspunsul în frecvență a funcției de transfer al senzației de forță specifică pentru sistemul otolitic este prezentat în Figura 6.1.

Tabel 6.1 Parametrii forței specifice pentru sistemul otolitic

	<b>X</b>	<b>Y</b>	<b>Z</b>
$k_{ot}$	0.4	0.4	0.4
$\tau_1$ [s]	5	5	5
$\tau_2$ [s]	0.016	0.016	0.016
$\tau_L$ [s]	10	10	10
$A_0$ [s <sup>-1</sup> ]	1/ $\tau_L$	1/ $\tau_L$	1/ $\tau_L$
$B_0$ [s <sup>-1</sup> ]	1/ $\tau_1$	1/ $\tau_1$	1/ $\tau_1$
$B_1$ [s <sup>-1</sup> ]	1/ $\tau_2$	1/ $\tau_2$	1/ $\tau_2$
$k_0$	$k_{ot}\tau_1\tau_2/\tau_L$	$k_{ot}\tau_1\tau_2/\tau_L$	$k_{ot}\tau_1\tau_2/\tau_L$
<b>Prag</b> [m/s <sup>2</sup> ]	0.17	0.17	0.28

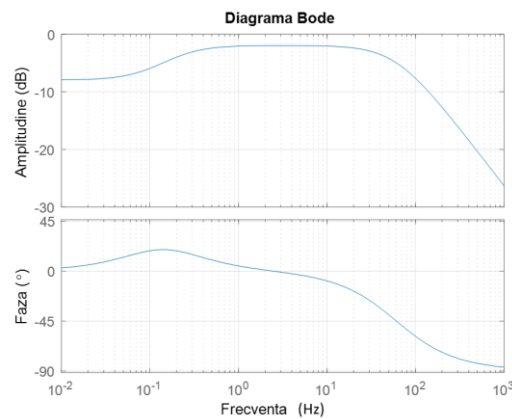


Figura 6.1 Răspunsul în frecvență pentru sistemul otolitic

În Tabel 6.2 sunt prezentați parametrii specifici pentru modelul de senzație a mișcării de rotație pentru sistemul semicircular și răspunsul în frecvență a funcției de transfer al senzației de rotație pentru canalul semicircular este prezentat în Figura 6.2.

Tabel 6.2 Parametrii specifici mișcării de rotație pentru sistemul semicircular

	<b>Ruliu (x)</b>	<b>Tangaj (y)</b>	<b>Girație (z)</b>
$k_{sc}$	28.6479	28.6479	35.8099
$\tau_1$ [s]	5.73	5.73	5.73
$\tau_2$ [s]	0.005	0.005	0.005
$\tau_L$ [s]	0.06	0.06	0.06
$\tau_a$ [s]	80	80	80
<b>Prag</b> [grad/s]	2.0	2.0	1.6

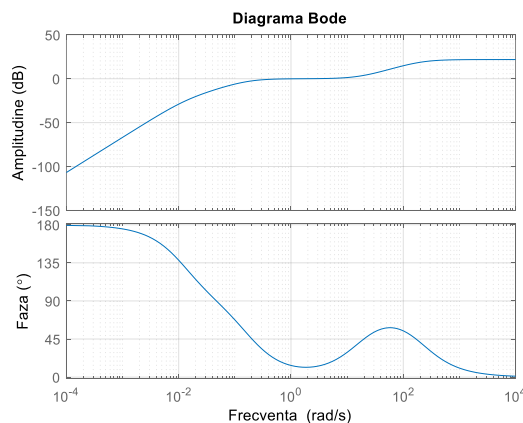


Figura 6.2 Răspunsul în frecvență pentru canalul semicircular

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

Una dintre principalele provocări în utilizarea AG este alegerea parametrilor AG adecvați, cum ar fi dimensiunea populației, rata de mutație și rata de încrucișare. Acești parametri definesc gradul de precizie a soluțiilor și viteza de convergență a algoritmului. Prin urmare, o setare adecvată a ratelor de încrucișare și de mutație este esențială pentru succesul rezultatelor oferite de aceasta. Acest lucru se realizează prin încercări și erori, deoarece ratele optime de încrucișare sau de mutație variază în funcție de problemele analizate. În experimentele prezentate ca parte din scenariile de testare a algoritmului, un total de 60 de cromozomi este considerat o dimensiune bună a populației pentru obținerea unor rezultate precise. Creșterea dimensiunii populației peste 60 de cromozomi în cazul studiat nu produce nicio îmbunătățire notabilă a soluției. În schimb, aceasta crește timpul de calcul și scade considerabil viteza de convergență. Pe de altă parte, deoarece soluția optimă depinde de inițializarea cromozomilor, valorile inițiale pentru fiecare cromozom sunt selectate pe baza soluțiilor anterioare ale soluției oferită de filtru de constrângere optimal bazat pe LQR. Astfel, se începe evoluția pe baza parametrilor din filtru de constrângere anterior. Pe baza cercetării realizate în literatura de specialitate se utilizează rata de încrucișare de 0.78, rata de mutație de 0.25 și un procent de 5% din dimensiunea populației pentru reproducerea elitelor și scalarea rangurilor.

În Figura 6.3 este prezentată schema evaluării ajustării algoritmului genetic pentru dezvoltarea filtrului de constrângere optimal bazat pe GA, implementat în mediul de simulare MATLAB / Simulink.

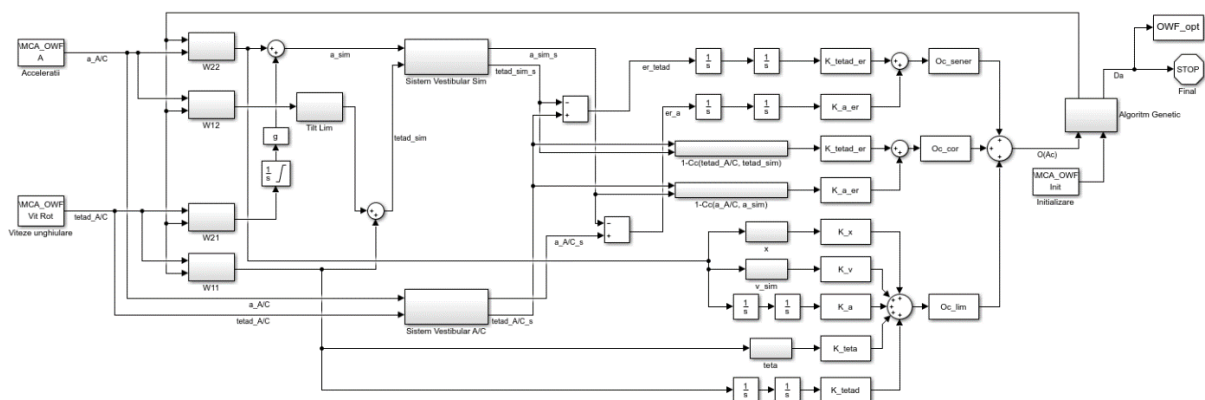


Figura 6.4 Schema evaluării ajustării AG pentru algoritmul filtru de constrângere optimal

În continuare vor fi prezentate rezultatele oferite de filtru de constrângere optimal bazat pe LQR și filtru de constrângere optimal modificat pe baza algoritmului genetic. Reprezentările grafice cuprind informații despre zbor și o comparație între rezultatele oferite de cele două filtre pentru accelerația liniară, mișcarea de rulu și de tangaj pentru diferite faze de zbor. Rezultatele vor fi prezentate pentru secvențe de timp scurte din fazele de zbor (croazieră și aterizare) de maxim 50 sec deoarece scenariul pentru cele șase zboruri a cuprins un tur de pistă 08R LROP ceea ce implică un timp de aproximativ 8 min per zbor, iar datele înregistrate au fost actualizate și stocate de 60 de ori pe secundă într-o bază de date. Din acest considerent, pentru o evidențiere a formei semnalului înregistrat de cele două filtre a fost aleasă varianta de reprezentare grafică pentru secvențe scurte de zbor, pentru zborurile (1-2). În Figura 6.5 sunt prezentați parametrii care descriu segmentul de zbor (latitudine – longitudine, altitudine, heading, numărul Mach) pentru secvența din faza de zbor de croazieră pentru primul scenariu

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

de zbor, unde forță gravitațională a înregistrat valoarea minimă și maximă astfel  $g_{min} = 0.3914$  și  $g_{max} = 2.2256$ .

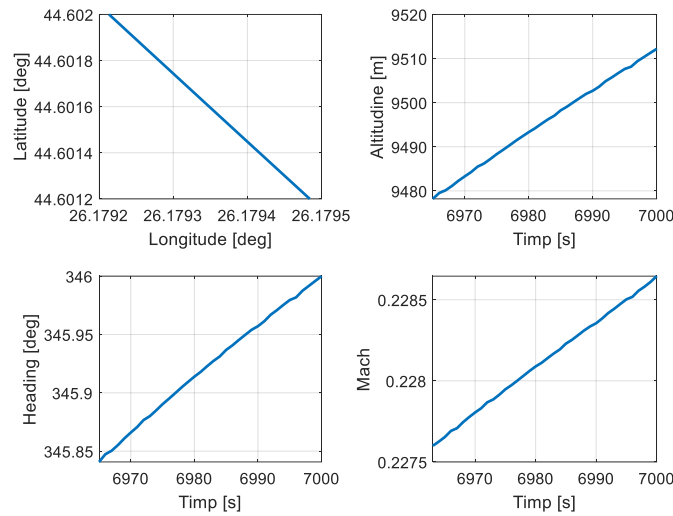


Figura 6.5 Scenariu de zbor (1) – Parametrii de zbor – secvență croazieră (1C)

În Figura 6.6, Figura 6.7 și Figura 6.8 se observă că semnalul înregistrat de algoritmul filtru de constrângere optimal modificat, bazat pe AG, aduce o corecție a formei semnalului pentru stările înregistrate de algoritmul inițial cu o îmbunătățire vizibilă pentru mișcarea de ruliu.

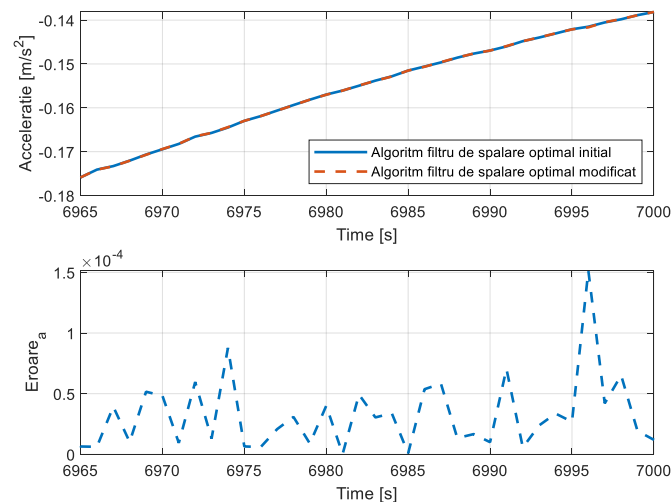


Figura 6.6 Accelația simțită – comparație între filtru de constrângere optimal inițial și filtru de constrângere optimal modificat (AG) (Scenariu Zbor 1C)

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

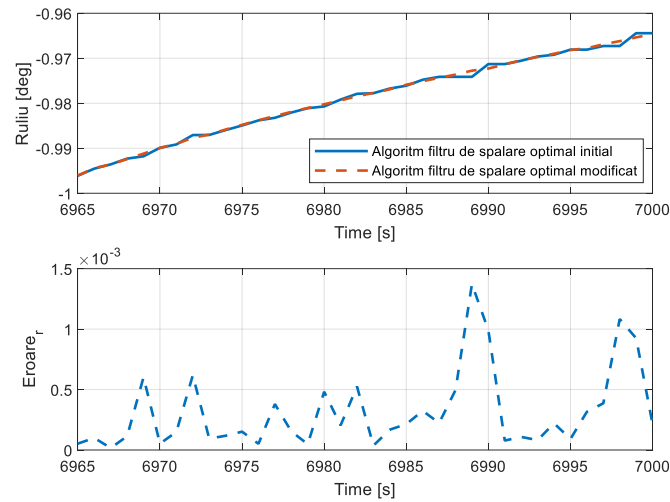


Figura 6.7 Unghi de ruluu – comparație între filtru de constrângere optimal inițial și filtru de constrângere optimal modificat (AG) (Scenariu Zbor 1C)

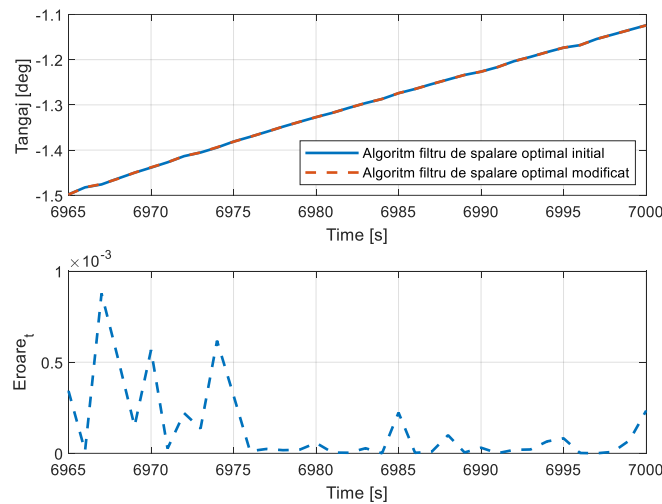


Figura 6.8 Unghi de tangaj – comparație între filtru de constrângere optimal inițial și filtru de constrângere optimal modificat (AG) (Scenariu Zbor 1C)

În Figura 6.9 sunt prezentați parametrii care descriu segmentul de zbor de aterizare (latitudine – longitudine, altitudine, heading, numărul Mach) pentru secvența aleasă pentru al doilea scenariu, unde forța gravitațională a înregistrat o valoare minimă de  $g_{min} = 0.5385$  și o valoare maximă de  $g_{max} = 1.5923$  pe durata zborului.

# Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

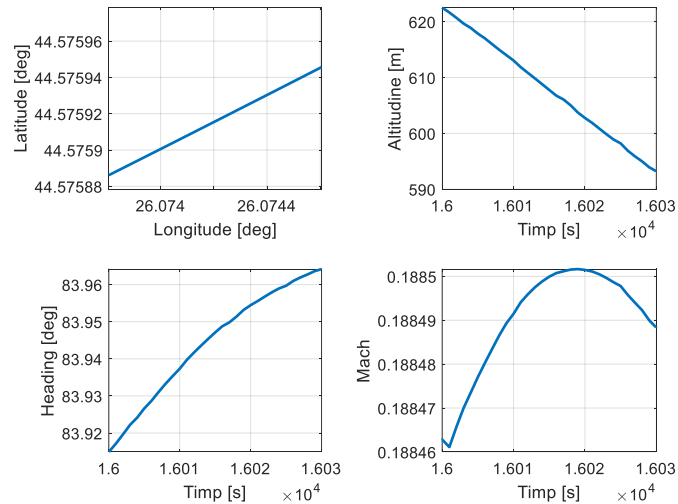


Figura 6.9 Scenariu de zbor (2) – Parametrii de zbor – secvență aterizare (2A)

Figura 6.10, Figura 6.11 și Figura 6.12 prezintă ieșirile filtrului de constrângere optimal modificat comparat cu algoritmul de mișcare OWF inițial fiind evidențiată eroarea medie între cei doi algoritmi ca fiind de ordinul  $10^{-4}$ .

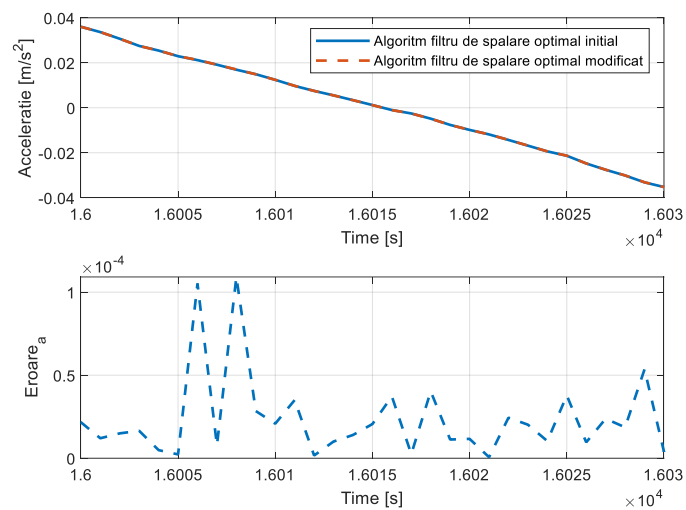


Figura 6.10 Accelerația simțită – comparație între filtru de constrângere optimal inițial și filtru de constrângere optimal modificat (AG) (Scenariu Zbor 2A)



## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

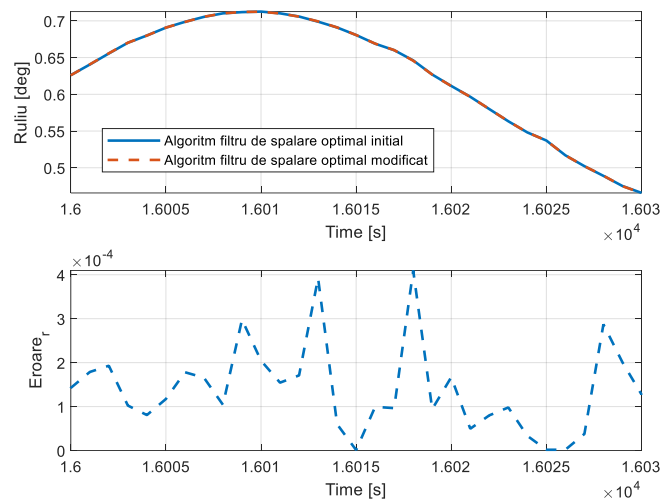


Figura 6.11 Unghi de rulu – comparație între filtru de constrângere optimal inițial și filtru de constrângere optimal modificat (AG) (Scenariu Zbor 2A)

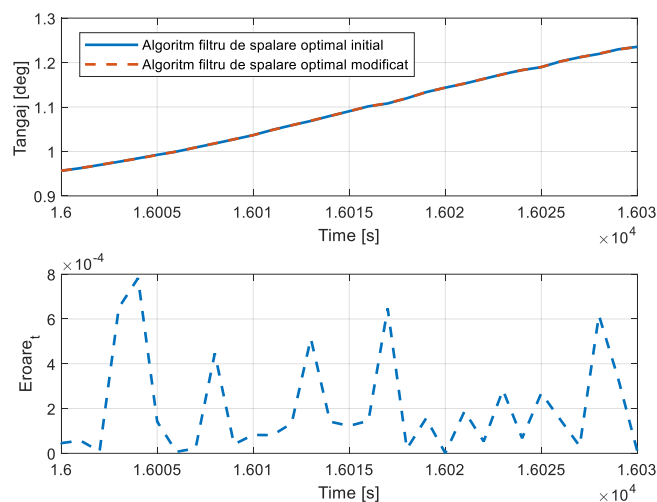


Figura 6.12 Unghi de tangaj – comparație între filtru de constrângere optimal inițial și filtru de constrângere optimal modificat (AG) (Scenariu Zbor 2A)

Rezultatele prezintă efectele referitoare la fidelitatea indicilor de mișcare. Din reprezentările grafice se observă că pentru mișcarea de tangaj și rulu modelele prezintă o continuitate aproape identică a formei semnalului, semnalul de ieșire este capabil să urmărească semnalul de referință al filtrului de constrângere modificat cu un grad de corelație semnificativ mai mare, media erorii înregistrate fiind de ordinul  $10^{-4}$ .

Algoritmul filtru de constrângere optimal modificat generează o mișcare mai bună cu o senzație mai realistă într-un mod mai eficient prin reducerea deplasării platformei. Ca urmare, simulatorul poate utiliza mai eficient spațiul suplimentar pentru alte mișcări posibile.

## 7. Concluzii și dezvoltări ulterioare

### 7.1. Concluzii generale

În prima parte a tezei de doctorat a fost realizat un studiu bazat pe cercetările existente în domeniul algoritmilor de indicare a mișcării pentru simulatoare de zbor. Pe baza rezultatelor prezentate în literatură de specialitate s-a realizat o analiză comparativă a patru tipuri de filtre de constrângere astfel: clasic, adaptiv, control optimal și control optimal robust. Au fost evidențiate avantajele și dezavantajele acestora pentru a se alege proiectarea pentru simulatorul de zbor a unui algoritm cât mai fezabil pentru platforma de mișcare de tip robot serial. Pe baza studiului s-a ales filtru de constrângere optimal deoarece acesta are incorporat modelul sistemului vestibular astfel fiind posibilă analiza erorii de percepție umană, iar reglarea algoritmului este accesibilă deoarece se realizează prin ajustarea ponderilor funcției de cost, ceea ce oferă un cadru favorabil pentru procedura de optimizare a algoritmului.

Totodată, a fost realizată modelarea matematică a sistemului de simulare dinamică pentru aplicații de tip pilot-in-the-loop, sistem utilizat ca platformă de testare pentru cele două configurații ale filtrului de constrângere optimal proiectat și implementat. Sistemul robotic de tip ABB IRB 7600-500 a fost utilizat ca platformă de mișcare în implementarea și testarea algoritmului de indicare a mișcării pentru simulatorul de zbor. A fost prezentată structura funcțională a sistemului de simulare dinamică și modul de interacțiune între componentele sistemului dinamic. Sistemele de referință pentru platforma de simulator au fost definite și reprezentate grafic.

Pentru testarea algoritmului de mișcare a fost necesară realizarea și simularea cinematicii inverse a platformei de mișcare de tip robot serial, care este reprezentată drept modul 4 conform structurii de simulare prezentată în Figura 3.2. Mișcarea robotului a fost controlată în diferite direcții folosind diverse combinații pentru unghiurile articulațiilor. Conceptul de cinematică directă și inversă a fost utilizat pentru a determina poziția efectorului final pentru unghiuri fixe ale articulațiilor și unghiurile articulațiilor pentru efectorul final fix. Problema cinematicii inverse a fost realizată utilizând algoritmul de optimizare Broyden–Fletcher–Goldfarb–Shanno (BFGS) în două medii de simulare diferite: MATLAB R2022b și Robot Studio, rezultatele simulării fiind prezentate în subcapitolul 3.4.2, putându-se confirma eficacitatea modelului prin deplasarea de-a lungul unei traiectorii predefinite a efectorului final al robotului serial. Rezultatul cinematicii inverse a fost, de asemenea, discutat pentru combinațiile de unghiuri articulare, fiind evidențiată diferența în configurația vectorului de poziție.

De asemenea, un modul esențial din algoritmul propus este scalarea și limitarea spațiului de lucru al simulatorului de zbor fiind redefinite limitele minime și maxime a articulațiilor platformei de mișcare pentru a păstra simulatorul în limitările și constrângerile sigure pe parcursul utilizării și testării algoritmilor propuși. Astfel pentru limitările hardware ale unghiurilor articulațiilor robotului au fost realizate modificări pentru articulația  $q_3$ ,  $q_4$  și  $q_5$ , astfel:  $[-180 +40]$ ,  $[-175 +175]$ , respectiv  $[-90 +90]$ .

Algoritmul de mișcare este responsabil pentru a transforma accelerația și viteza unghiulară a unui vehicul simulat cu scopul de a reproduce mișcări de înaltă fidelitate în limitele fizice ale simulatorului. În lucrarea de doctorat, filtru de constrângere optimal a fost proiectat pe baza

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

metodei LQR care ia în considerare sistemul vestibular și mișcarea simulatorului pentru a reduce eroarea de percepție umană între simulator și avionul real într-un mod cât mai eficient.

Datorită limitărilor fizice ale sistemului robotic serial de tip ABB IRB 7600-500 pentru canalul translațional și rotațional a fost aplicată o limitare. Totodată pentru semnalele vectorului de stare ( $u_{A/C}$ ) utilizat ca intrare în modelul algoritmului a fost aplicată o scalare pentru modificarea amplitudinii semnalului în mod uniform pe toate frecvențele. Un polinom de ordinul trei a fost utilizat și implementat în schema generală a algoritmului filtru de constrângere optimal. Unitățile de scalare și de limitare joacă un rol important în menținerea mișcărilor în limitele fizice ale simulatorului de zbor, fiind utilizate împreună cu filtru de constrângere optimal pentru a reduce magnitudinea semnalelor de mișcare de translație și de rotație în mod uniform pe toate frecvențele din algoritm, astfel încât să se diminueze efectele datorate limitărilor spațiului de lucru în reproducerea mișcărilor simulatorului și să se îmbunătățească realismul senzației de mișcare.

În sub-capitolul 5.3. a fost realizată modelarea matematică a sistemului vestibular care este un modul esențial din structura algoritmului de mișcare de tip filtru de constrângere optimal.

Totodată, a fost ales sistemul de referință cu centrul de rotație la nivelul capului pilotului pentru a elimina mișcările false și senzația de cuplare încrucișată între rotația simulatorului și mișcarea de translație a capului pilotului, precum și pentru a reduce deplasarea în spațiul de lucru al platformei de mișcare.

Pentru a îmbunătăți filtrul de constrângere optimal bazat pe LQR, a fost propusă o nouă strategie bazată pe algoritmi evolutivi. Principalul obiectiv fiind regenerarea unui semnal care poate urmări îndeaproape semnalul de referință, poate să evite indicii falși de mișcare și își poate îmbunătăți forma. O serie de criterii au fost luate în considerare în funcția de cost a algoritmului genetic, precum eroarea dintre senzația de mișcare înregistrată de pilotul real și cea simulată; impunerea unor limitări dinamice ale simulatorului; stabilirea limitatorului de prag uman în coordonarea înclinării; derivata erorii de senzație umană; și coeficientul de corelație încrucișată.

Rezultatele arată că metoda optimă propusă bazată pe AG poate gestiona toate aceste aspecte într-un mod cuprinzător. Algoritmii de indicare a mișcării optimi bazat pe AG au fost implementați utilizând pachetul software MATLAB/ Simulink.

Rezultatele obținute arată superioritatea filtrului de constrângere optimal propus bazat pe AG datorită performanțelor sale mai bune, a senzației umane îmbunătățite, a factorului crescut de urmărire a formei și a deplasării reduse. Deoarece s-a obținut un echilibru între eroarea de senzație umană și urmărirea formei, filtrul de constrângere optimal propus, bazat pe AG, generează o mișcare mai bună și poate oferi o senzație mai realistă pilotului din simulator într-un mod mai eficient. De asemenea, filtrul de constrângere optimal implementat și testat poate utiliza spațiul de lucru mai eficient, permițând o mai mare libertate de acțiune pentru includerea altor mișcări posibile. Rezultatele au arătat că algoritmul este fiabil, robust și eficient în această aplicație pentru simulatorul de zbor bazat pe platformă de mișcare robot serial de tip ABB IRB 7600-500.

Avantajul acestui studiu este că algoritmul propus poate fi utilizat ca instrument pentru orice alt tip de simulator cu diferite limitări fizice sau pentru același simulator în alte configurații

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

fizice, fără a fi nevoie de un efort ridicat pentru reglarea parametrilor specifici algoritmului de indicare a mișcării.

### 7.2. Contribuții proprii

În lucrarea de doctorat a fost proiectat, implementat și testat un algoritm de indicare a mișcării, filtru de constrângere optimal bazat pe LQR și o versiune îmbunătățită a acestuia bazată pe algoritmi genetici, fiind prima utilizare a acestui tip de algoritm pentru simulatoare de zbor bazate pe platforme de mișcare de tip robot serial ABB IRB 7600-500. Conform studiului realizat în capitolul 2 până acum acest algoritm a fost proiectat, implementat și testat doar pentru simulatoare bazate pe platforme de mișcare de tip robot paralel (platforme de tip Stewart).

Limitarea și scalarea platformei de mișcare de tip ABB IRB 7600-500 este un alt aspect de noutate pentru sistemul robotic serial utilizat în prezentul studiu deoarece implică o procedură specifică aplicației pentru simulatorul de zbor.

Una dintre cele mai cunoscute provocări în proiectarea și implementarea algoritmilor de indicare a mișcării este alocarea valorilor articulațiilor pentru a controla percepția umană și pentru a realiza o mișcare cât mai fidelă mișcării reale. Algoritmul filtru de constrângere optimal implementat diminuează eroarea de percepție umană și pregătește o mișcare în spațiul de lucru al simulatorului de zbor. Definierea funcției obiectiv este descrisă de două criterii: minimizarea spațiului de lucru al simulatorului de zbor și minimizarea erorii de percepție umană, fiind luate în considerare două manevre diferite (rului și tangaj). Rezultatele prezentate în capitolul 6 arată că mișcarea se desfășoară în spațiul de lucru al simulatorului, iar eroare de percepție este foarte scăzută, în condițiile în care corelațiile percepției umane înainte și după aplicarea filtrului de constrângere propus sunt mai mari de 85% pentru toate manevrele studiate, adică percepția umană nu se modifică în mod intens atunci când se aplică filtru de constrângere.

Totodată, în cadrul algoritmului filtru de constrângere optimal, în funcția de cost a fost utilizată maximizarea coeficientului de corelație, definit ca fiind coeficientul de corelație de moment Pearson, pentru a crește gradul de urmărire a formei semnalului de referință în semnalul produs. Permitea ca semnalul regenerat să urmeze forma semnalului de la pilotul real este un criteriu important, pe lângă diminuarea erorii de senzație umană. Acest aspect semnificativ nu a fost luat în considerare în funcția de cost a algoritmilor de indicare a mișcării implementați până acum.

Filtrul de constrângere propus reduce simultan accelerația reală în spațiul de lucru al simulatorului și eroarea sistemului de percepție. Metoda propusă este capabilă să producă cu acuratețe mișcări ale platformei de simulator de zbor cu o fidelitate mai mare și o eficiență mai mare în utilizarea platformei robotice, fără a încălca limitele fizice ale simulatorului.

Un alt aspect de noutate este îmbunătățirea funcționalității algoritmului de indicare a mișcării OWF utilizând algoritmul genetic pentru configurația de simulator cu șase grade de libertate.

### 7.3. Perspective de dezvoltare a activității de cercetare

Pentru posibile dezvoltări viitoare în vederea îmbunătățirii algoritmului de indicare a mișcării, pot fi implementate și testate alte metode de optimizare pentru a crește fidelitatea mișcării în diferite scenarii de zbor și condiții de funcționare a platformei de simulator de zbor.

Totodată sunt necesare cercetări suplimentare pentru a înțelege mai bine evaluările subiective ale piloților de teste. De asemenea, experimente subiective suplimentare care să se concentreze asupra efectului indicilor falși și a erorilor asupra fidelității percepției mișcării pot conduce la îmbunătățirea aplicabilității algoritmului.

Un alt aspect important de luat în considerare în viitor va fi selectarea optimă a poziției inițiale a simulatorului de zbor pentru a îmbunătăți utilizarea spațiului de lucru al simulatorului pentru o anumită sarcină de simulare.

Ca cercetări viitoare, metoda propusă implementată în mediul de simulare MATLAB / Simulink ca program de calcul dezvoltat pentru optimizarea algoritmilor de indicare a mișcării pentru simulatoare de zbor poate fi utilizată pentru a evalua și alte tipuri de algoritmi de indicare a mișcării.

Metodele de calcul evolutiv pot fi, de asemenea, utilizate pentru a proiecta unitatea de scalare neliniară. Aceasta este o direcție pentru cercetările viitoare, luând în considerare să se investigheze și să se proiecteze o unitate de scalare neliniară bazată pe intervale de predicție. Totodată, efectuarea unor teste suplimentare cu ajutorul simulatorului de zbor pentru analiza rezultatelor oferite de metoda propusă face parte din faze viitoare ale acestei cercetări.

Pentru o utilizare mai ușoară și eficientă a programului de calcul dezvoltat pentru optimizarea algoritmilor de indicare a mișcării este importantă și dezvoltarea unei interfețe grafice cu o suită de intrări specifice fiecărui modul din algoritmul de indicare a mișcării, și totodată cu posibilitatea comparării rezultatelor oferite de diferiți algoritmi de mișcare pentru o selectare adecvată în diferite scenarii de testare și configurații ale simulatorului de zbor.

**Bibliografie** (selectivă)

- [1] Brain C J., Clayton T D., Ward N J., „Organizing safety related aircraft simulation,” *Test and Evaluation International Aerospace Forum* (2), London, 25-27 June, 1996.
- [2] Allerton D.J., „The impact of flight simulation in aerospace,” *The Aeronautical Journal*, vol. 114, nr. 1162, pp. 747 - 756, December 2010.
- [3] Olivotto C., „Simulators and Training Systems Somewhere in East - Realities, Demands and the Future,” în *3rd International Training Equipment Conference and Exhibition Proceedings*, Luxemburg, April 7-11, 1992.
- [4] Olivotto C., „The Training Equipment in a Pilot-School Modernization - Flight Simulation, Technologies, Capabilities and Benefits,” în *London Royal Aeronautical Society*, May 17-18 1995.
- [5] Chira A., Dumitrescu A., Moisoiu C.S., Tanasa C., „Human Performance Envelope Model Study Using Pilot’s Measured Parameters,” *INCAS Bulletin*, vol. 12, nr. 4, pp. 49-61, 2020.
- [6] Bellmann T., Otter M., Hirzinger G., „The DLR robot motion simulator part I: design and setup,” în *IEEE International Conference on Robotics and Automation*, Shanghai, 2011.
- [7] Bellmann T., Otter M., Hirzinger G., „The DLR robot motion simulator part II: optimization based path-planning,” în *IEEE International Conference on Robotics and Automation*, Shanghai, 2011.
- [8] Teufel HJ., Nusseck HG., Beykirch KA., Butler JS., Kerger M., Bühlhoff HH., „MPI motion simulator: development and analysis of a novel motion simulator,” în *AIAA Modeling and Simulation Technologies Conference and Exhibit*, South Carolina, 2007.
- [9] Chira A.I., Bălașa R.I., Bîlu M.C., Iordache C., Moisoiu C.S., Munteanu C., „Design and Analysis of a Flight Simulator based on a Robotic Motion Platform,” în *8th International Workshop on Numerical Modelling in Aerospace Sciences*, Bucharest, 25 May, 2022.
- [10] Chira A.I., Bălașa R.I., Bîlu M.C., Iordache C., „Workspace Optimization of A Flight Simulator based on a Robotic Motion Platform,” în *8th International Workshop on Numerical Modelling in Aerospace Sciences*, Bucharest, 25 May, 2022.
- [11] Chira A.I., Bîlu M.C., Bălașa R.I., Iordache C., „Enhancing Realism in Robotic Flight Simulators: a case study on the implementation analysis of a washout filter,” în *34th DAAAM International Symposium on Intelligent Manufacturing and Automation*, Vienna, October, 2023.
- [12] Bogos S., Chira A., Udrea I.P., Sandu V., Draghici T., „Comprehensive Analysis of Aircraft Dynamics Stability with SCSim: Integration and Assessment,” *INCAS Bulletin*, vol. 15, nr. 4, pp. 41-51, 2023.
- [13] Grant P. R. , Reid L. D. , „Motion washout filter tuning: Rules and requirements,” *Journal Aircraft*, vol. 34, nr. 2, pp. 145-151, 1997.
- [14] Sivan R., Ish-Shalom J., Huang J.-K., „An optimal control approach to the design of moving flight simulators,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 12, nr. 6, pp. 818-827, 1982.
- [15] Parrish R. V., Dieudonne J. E., Bowles R. L., & Martin Jr D. J., „Coordinated adaptive washout for motion simulators,” *Journal of Aircraft*, vol. 12, nr. 1, pp. 44-50, 1975.

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

- [16] Asadi H., Mohamed S., Lim C. P., Nahavandi S., „Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1-17, 2016.
- [17] Chen S. H., Fu L.-C., „An optimal washout filter design with fuzzy compensation for a motion platform,” *18th IFAC World Congress, IFAC Proceedings*, vol. 44, nr. 1, pp. 8433-8438, 2011.
- [18] Gharib M. R., „Comparison of robust optimal QFT controller with TFC and MFC controller in a multi-input multioutput system,” *Reports in Mechanical Engineering*, vol. 1, nr. 1, pp. 151-161, 2020.
- [19] Huang C., Fu L., „Human Vestibular Based (HVB) Senseless maneuver optimal washout filter design for VR-based motion simulator,” *IEEE International Conference on Systems, Man and Cybernetics*, pp. 4451-4458, 2006.
- [20] Kong X.-T., Zhu Y.-C., Di Y.-Q., Cui H.-H., „Methods to determine optimal washout position for single and multioccupant motion simulator,” *Cybernetics and Information Technologies*, vol. 16, nr. 1, pp. 173-187, 2016.
- [21] Arioui H., Nehaoua L., Amouri A., „Classic and adaptive washout comparison for a low cost driving simulator,” *Proceedings of the 2005 IEEE International Symposium on Mediterrean Conference on Control and Automation*, pp. 586-591, 2005.
- [22] Chen S. H., Fu L. C., „An optimal washout filter design for a motion platform with senseless and angular scaling maneuvers,” *IFAC Proceedings*, vol. 44, nr. 1, pp. 8433-8438, 2010.
- [23] Wang X. L., Li L., Zhang W. H., „Research on fuzzy adaptive washout algorithm of train driving simulator,” *Tiedao Xuebao/Journal of the China Railway Society*, vol. 32, nr. 2, pp. 31-36, 2010.
- [24] Yang Y., Huang Q.-T., Han J.-W., „Adaptive washout algorithm based on the parallel mechanism motion range,” *Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics*, vol. 32, nr. 12, pp. 2716-2720, 2010.
- [25] Moavenian M., Gharib M. R., Daneshvar A., Alimardani S., „Control of human hand considering uncertainties,” *IEEE International Conference on Advanced Mechatronic Systems*, pp. 17-22, 2011.
- [26] Kim K. D., Kim M. S., Moon Y. G., Lee M. C., „Application of vehicle driving simulator using new washout algorithm and robust control,” *SICE-ICASE International Joint Conference*, pp. 2121-2126, 2006.
- [27] Salehi Kolahi M. R., Gharib M. R., Koochi A., „Design of a robust control scheme for path tracking and beyond pull-in stabilization of micro/nano-positioners in the presence of Casimir force and external disturbances,” *Archive of Applied Mechanics*, vol. 91, nr. 10, pp. 4191-4204, 2021.
- [28] Nehaoua L., Amouri A., Arioui H., „Classic and adaptive washout comparison for a low cost driving simulator,” în *IEEE International Symp. Mediterranean Conference Control Automation Intell. Control*, Limassol, 2005.
- [29] Nehaoua L., Arioui H., Mohellebi H., Espie S., „Restitution movement for a low cost driving simulator,” în *Proceeding American Control Conference*, Minneapolis, 2006.
- [30] Nehaoua L., Mohellebi H., Amouri A., Arioui H., Espié S., Kheddar A., „Design and Control of a Small-Clearance Driving Simulator,” *IEEE Transactions on Vehicular Technology*, vol. 57, nr. (2), pp. 736-746, 2008.

## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

- [31] Arioui H., Hima S., Nehaoua L., Bertin R. J. V., Espié S. , „From design to experiments of a 2-DOF vehicle driving simulator,” *IEEE Transactions on Vehicular Technology* , vol. 60, nr. (2), pp. 357-368, 2011.
- [32] Reid L.D., Nahon M.A. , „Flight simulation motion-base drive algorithms: Part 1. Developing and testing the equations,” Technical Report 296, Toronto, 1985.
- [33] Reid L.D., Nahon M.A., „Flight simulation motion-base drive algorithms: Part 2. Selecting the system parameters,” Technical Report 307, Toronto, 1986.
- [34] Reid L.D., Nahon M.A., „Flight simulation motion-base drive algorithms. Part 3: Pilot evaluations,” Technical Report 319, Toronto, 1986.
- [35] Reid L. D. , Nahon M. A., „Response of airline pilots to variations in flight simulator motion algorithms,” *Journal Aircraft*, vol. 25, nr. 7, pp. 639-646, 1988.
- [36] Nahon M. A. , Reid L. D. , „Simulator motion-drive algorithms - A designer's perspective,” *Journal of Guidance, Control, and Dynamics*, vol. 13, nr. 2, pp. 356-362, 1989.
- [37] Zacharias G. L. , „Motion Cue Models for Pilot-Vehicle Analysis,” AMRL-TR-78-2, Ohio, USA, 1978.
- [38] Hosman R. J. A. W., J. C. van der Vaart, „Vestibular models and thresholds of motion perception: Results of tests in a flight simulator,” Delft, The Netherlands, 1978.
- [39] Telban R. J., Wu W., Cardullo F. M., „Motion cueing algorithm development: Initial investigation and redesign of the algorithms,” Technical Report NASA/CR-2000-209863, Hampton, VA, USA, 2000.
- [40] Nahon M.A., Reid L.D., „Simulator motion-drive algorithms: a designer's perspective,” *Journal of Guidance, Control and Dynamics*, vol. 13, nr. 2, pp. 356-362, 1990.
- [41] Reid L.D., Nahon M.A., Kirdeikis J. , „Adaptive simulator motion software with supervisory control,” *Journal of Guidance, Control and*, vol. 15, nr. 2, pp. 376-383, 1992.
- [42] FAA - Federal Aviation Administration, „Flight test guide for certification of transport category airplanes,” AC 25-7C, 2012. [Interactiv]. Available: [http://www.faa.gov/documentLibrary/media/Advisory\\_Circular/AC\\_25-7C.pdf](http://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_25-7C.pdf). [Accesat 12 September 2022].
- [43] Garrett N.J.I., Best M.C., „Driving simulator motion cueing algorithms – a survey of the state of the art,” *10th International Symposium on Advanced Vehicle Control*, 2010.
- [44] Chira A.I., Afloare A.I., Apostolescu A., Munteanu C.E., „Inverse kinematic solution of a 6 DoF serial manipulator,” în *Proceedings of the 30th DAAAM International Symposium*, Vienna, Austria, 2019.
- [45] Denavit J., Hartenberg R. S., „A kinematic notation for lower-pair mechanisms based on matrices,” în *Journal of Applied Mechanics*, 1955.
- [46] MATLAB Robotics Toolbox, [Interactiv]. Available: <https://www.mathworks.com/help/robotics/>. [Accesat 14 09 2022].
- [47] Corke P., „Robotics, Vision & Control - ROBOTICS Toolbox,” [Interactiv]. Available: <https://petercorke.com/toolboxes/robotics-toolbox/#>. [Accesat 14 09 2022].
- [48] ABB Robotics - Product specification IRB 7600, ABB, [Interactiv]. Available: <https://new.abb.com/products/robotics/robots/articulated-robots/irb-7600>. [Accesat 12 09 2022].
- [49] Rubio J.D., Aquino V. , Figueroa M. , „Inverse kinematics of a mobile robot,” *Neural Computer & Applications*, vol. 23, nr. 1, pp. 187-194, 2013.



## Contribuții asupra algoritmilor de indicare a mișcării pentru simulatoare de zbor

- [50] ABB Robotics - Robot Studio, ABB, [Interactiv]. Available: <https://new.abb.com/products/robotics/robotstudio>. [Accesat 14 09 2022].
- [51] ABB - Rapid Overview Manual, [Interactiv]. Available: <https://new.abb.com/products/robotics/robotstudio>. [Accesat 14 09 2022].
- [52] Schreiber G. , „Control for redundant robot systems: User and task-oriented use of redundancy,” 2004.
- [53] Van Egmond A.A., Groen J. J., Jongkees L. B. W., „The Mechanics of the Semicircular Canal,” *Journal of Physiology*, vol. 110, pp. 1-17, 1949.
- [54] Howard I.P., „Handbook of Perception and Human Performance, The Vestibular System,” 1986.
- [55] Meiry J.L., „The Vestibular System and Human Dynamic Space Orientation,” Sc.D. Thesis, Cambridge, MA, 1965.
- [56] Telban R. J., Cardullo F. M., „Motion cueing algorithm development: Human-centered linear and nonlinear approaches,” NASA Langley Research, Technical Report CR-2005-213747, Hampton, VA, USA, 2005.
- [57] Fleming P.J., Pyrshouse R.C., „Evolutionary algorithms in control systems engineering: A survey,” în *Control Engineering Practice*, 2002.
- [58] Yang X.S., „Chapter 5 Genetic Algorithms,” în *Nature-Inspired Optimization Algorithms*, 2014.
- [59] Johar F.M., „A review of genetic algorithms and parallel genetic algorithms on graphics processing unit (GPU),” în *Proc. IEEE Int. Conf. Control Syst. Comput. Eng. (ICCSCE)*, Penang, 2013.
- [60] Schmidt S.F., Conrad B., „Motion drive signals for piloted flight simulators,” *CR-1601 NASA Contractor Report*, 1970.
- [61] Reid L.D., Nahon M.A., „Flight simulation motion-base drive algorithms part II: selecting the system parameters,” *N307, UTIAS Report*, 1986a.
- [62] Reid L.D., Nahon M.A., „Flight simulation motion-base drive algorithms part III: pilot evaluations,” *N319, UTIAS Report*, 1986b.